

# Reconeixement de noms per a l'ajut a la correcció d'exàmens

*Marcel Arroyo Pinar*

Director: Josep Llosa  
Codirector: Carlos Álvarez  
Especialització: Enginyeria de Computadors



Universitat Politècnica de Catalunya

# Abstract

The correction of exams in the Barcelona School of Informatics has become a heavy task due to the increase of students enrolled annually. The teachers of the subject on Computer Architecture have detected that a great amount of time is wasted, classifying and ordering the exams to update the students grades.

To be able to automate some of the tasks that involve a considerable waste of time during the correction process, it would be necessary to create a tool capable of recognizing whose is the exam of a scanned document.

This project aims to make the association of these digitized documents with their respective students. For this purpose, a tool capable of recognizing letters and digits written by hand has been implemented, which will allow us to automate this association process. Neural networks have been used to create this recognition tool.

Neural networks, as the name suggests, try to simulate human behavior but with a completely different strategic approach, repeated training. In this case, what you want is to recognize alphanumeric characters, neural networks are based on collecting the largest number of images of handwritten characters to later develop a system that can learn from these initial data.

# Resum

La correcció d'exàmens a la Facultat d'Informàtica de Barcelona s'ha convertit en una tasca pesada degut a l'increment d'alumnes matriculats anualment. El professorat de l'assignatura d'Arquitectura de Computadors ha detectat que es perd una gran quantitat de temps classificant i ordenant els exàmens per a actualitzar les notes de l'alumnat.

Per tal de poder automatitzar algunes de les tasques que impliquen una pèrdua considerable de temps durant el procés de correcció, seria necessària la creació d'una eina capaç de reconèixer de qui és l'examen que conté un document digitalitzat.

Aquest projecte pretén realitzar l'associació d'aquests documents digitalitzats amb el seu respectiu alumne. Per a fer-ho s'ha implementat una eina capaç de reconèixer caràcters alfabètics i numèrics escrits a mà, que ens permetrà automatitzar aquest procés d'associació.

Per tal de crear aquesta eina de reconeixement s'han usat xarxes neuronals. Les xarxes neuronals com el seu nom indica intenten simular el comportament humà però amb un enfoc estratègic completament diferent, l'entrenament reiterat. En aquest cas que el que es vol es reconèixer caràcters alfanumèrics, les xarxes neuronals es basen en recollir el major nombre d'imatges de caràcters escrits a mà per tal de posteriorment desenvolupar un sistema que pugui aprendre d'aquestes dades inicials.

# Resumen

La corrección de exámenes en la Facultad de Informática de Barcelona se ha convertido en una tarea pesada debido al incremento de alumnos matriculados anualmente. El profesorado de la asignatura de Arquitectura de Computadores ha detectado que se pierde una gran cantidad de tiempo clasificando y ordenando los exámenes para actualizar las notas del alumnado.

Para poder automatizar algunas de las tareas que implican una pérdida considerable de tiempo durante el proceso de corrección, sería necesaria la creación de una herramienta capaz de reconocer de quién es el examen que contiene un documento digitalizado.

Este proyecto pretende realizar la asociación de estos documentos digitalizados con su respectivo alumno. Para ello se ha implementado una herramienta capaz de reconocer letras y dígitos escritos a mano, que nos permitirá automatizar este proceso de asociación.

Para crear esta herramienta de reconocimiento se han usado redes neuronales. Las redes neuronales como su nombre indica intentan simular el comportamiento humano, pero con un enfoque estratégico completamente diferente, el entrenamiento reiterado. En este caso que lo que se quiere es reconocer caracteres alfanuméricos, las redes neuronales se basan en recoger el mayor número de imágenes de caracteres escritos a mano para posteriormente desarrollar un sistema que pueda aprender de estos datos iniciales.

# Agraïments

Gràcies a en Josep Llosa i a en Carlos Alvarez per donar-me l'oportunitat de dur a terme aquest projecte. Gràcies per tots els consells i recomanacions.

Gràcies a tots els amics que m'han sofert durant aquests mesos. Gràcies per tot el suport mostrat.

Gràcies als meus pares per donar-me la possibilitat de cursar aquest grau. Gràcies per tot el suport i confiança mostrada en aquests 5 anys.

A tots vosaltres, mil gràcies!

# Índex

1 Introducció	13
1.1 Context	13
1.2 Actors	14
1.2.1 Desenvolupador	14
1.2.2 Director i codirector	14
1.2.3 Usuaris	15
1.2.4 Col·laboradors del projecte	15
1.2.5 Beneficiaris	15
1.3 Estat de l'art	15
1.3.1 Evolució de les xarxes neuronals	15
1.3.2 Eines de reconeixement de textos escrits a mà	18
1.4 Identificació de lleis i regulacions	19
1.4.1 Llicenciament	19
1.4.2 Propietat intel·lectual	19
1.4.3 Protecció de dades	19
1.5 Abast del projecte	20
1.6 Formulació del problema	20
1.7 Metodologia	21
1.7.1 Mètodes de treball	21
1.7.2 Eines de desenvolupament	21
1.7.3 Eines de seguiment	22
1.7.4 Mètode de validació	22
1.8 Integració de conceptes	22
1.8.1 Processat d'imatges	22
1.8.2 Machine learning	23
1.8.3 Correcció d'errors	23
1.9 Obstacles i riscos del projecte	23
1.9.1 Dades estandarditzades	23
1.9.2 Temps de computació	23
1.9.3 Complexitat de les capes interiors	24
2 Planificació temporal	25

2.1 Descripció de les tasques	25
2.1.1 Gestió del projecte	25
2.1.2 Adquirir coneixements sobre xarxes neuronals	25
2.1.3 Selecció del framework de Deep Learning	25
2.1.4 Elaboració de la xarxa neuronal	26
2.1.5 Entrenament de la xarxa neuronal	26
2.1.6 Avaluació de la xarxa neuronal	26
2.1.7 Tractament d'errors	27
2.1.8 Etapa Final	27
2.2 Estimació del temps per a cada tasca	28
2.3 Diagrama de Gantt	29
2.4 Recursos	30
2.4.1 Recursos Humans	30
2.4.2 Recursos Hardware	30
2.4.3 Recursos Software	30
2.5 Modificacions respecte la planificació inicial	30
2.5.1 Desviacions	30
2.5.2 Pla d'actuació	31
2.6 Pressupost	32
2.6.1 Identificació i estimació dels costos	32
2.6.2 Control de gestió	35
2.7 Sostenibilitat i compromís social	35
2.7.1 Dimensió econòmica	36
2.7.2 Dimensió ambiental	36
2.7.3 Dimensió social	37
3 Disseny de l'eina	38
3.1 Integració del desenvolupament en el context de l'aplicació	38
3.2 Criteris de disseny	38
3.2.1 Processat d'imatge	39
3.2.2 Machine learning	39
3.2.3 Correcció d'errors	41
3.3 Disseny de l'eina	41
3.3.1 Lectura	42
3.3.2 Identificació de la graella	42

3.3.3 Identificació dels camps i les caselles	43
3.3.4 Tractament de la imatge	43
3.3.5 Entrenament de la xarxa neuronal	43
3.3.6 Predicció de caràcters	44
3.3.7 Reconstrucció de les dades personals mitjançant les prediccions	45
3.3.8 Correcció d'errors i assignació d'exàmens	45
3.3.9 Sortida	47
4 Implementació	48
4.1 Digitalització d'imatges	48
4.2 Creació d'un document digitalitzat per a cada examen	50
4.3 Conversió dels documents PDF a imatges JPG	50
4.4 Identificació de les coordenades de la graella	50
4.4.1 Suavitització de la imatge	51
4.4.2 Binarització i retall de la imatge	52
4.4.3 Identificació dels contorns de la graella	52
4.4.4 Redimensionament de la imatge	53
4.4.5 Identificació dels contorns de cada camp	54
4.4.6 Identificació dels contorns de cada casella	54
4.5 Tractament de la imatge final	55
4.5.1 Retallat de la imatge	55
4.5.2 Redimensionament equitatiu de la imatge	55
4.5.3 Correcció del color	56
4.5.4 Binarització de colors	56
4.5.5 Identificació de caselles buides	57
4.6 Arquitectura de la xarxa neuronal	57
4.6.1 Capa d'entrada	57
4.6.2 Capa intermèdia o amagada	58
4.6.3 Capa de sortida	60
4.6.4 Models proposats	60
4.7 Entrenament de la xarxa neuronal	62
4.7.1 MNIST	62
4.7.2 EMNIST	62
4.7.3 Augmentació de dades	63
4.8 Avaluació de la xarxa neuronal	64



4.9 Estructura de la predicció	64
4.10 Algoritmes d'identificació	64
4.10.1 Associació directa	65
4.10.2 Associació directa amb un error	65
4.10.3 Associació d'exàmens que presenten més d'un error	65
4.11 Entrada i sortida	66
4.11.1 Entrada	66
4.11.2 Sortida	68
5 Resultats	69
6 Conclusions i treball futur	75
6.1 Conclusions	75
6.2 Treball futur	75
Referències	78
Apèndix	80

## Llista de figures

Figura 1: Model d'una neurona artificial .....	16
Figura 2: Model d'una neurona artificial .....	17
Figura 3: Gràfica de la funció sigmoide .....	17
Figura 4: Declaració jurada per a l'ús de dades personals .....	19
Figura 5: Diagrama de Gantt.....	29
Figura 6: Diagrama de flux corresponent a l'etapa de lectura .....	42
Figura 7: Diagrama de flux de la identificació de la graella .....	42
Figura 8: Diagrama de flux del tractament de la imatge .....	43
Figura 9: Diagrama de flux de l'entrenament de la xarxa neuronal.....	44
Figura 10: Diagrama de flux de la predicció de caràcters.....	45
Figura 11: Diagrama de flux de l'algoritme d'assignació directa .....	46
Figura 12: Diagrama de flux de l'algoritme d'assignació directa amb un error .....	46
Figura 13: Diagrama de flux de l'algoritme d'assignació amb més d'un error .....	47
Figura 14: Format d'examen corresponent al 2n Control del Curs 2017-2018 Q1.....	48
Figura 15: Format d'examen corresponent al 1er Control del Curs 2017-2018 Q2.....	49
Figura 16: Graella corresponent a l'antic format emprat .....	49
Figura 17: Graella corresponent a l'actual format emprat.....	49
Figura 18: Graella usada en els exàmens .....	51
Figura 19: Imatge sense cap tipus d'efecte.....	51
Figura 20: Imatge suavitzada mitjançant el desenfocament gaussià.....	51
Figura 21: Graella binaritzada.....	52
Figura 22: Línies horitzontals de la graella .....	52
Figura 23: Línies verticals de la graella .....	52
Figura 24: Graella simplificada.....	53
Figura 25: Vèrtexs de la graella simplificada.....	53
Figura 26: Interpolació bilineal .....	53
Figura 27: Camp on apareix el cognom de l'alumne .....	54
Figura 28: Camp on apareix el nom de l'alumne.....	54
Figura 29: Camp on apareix el DNI de l'alumne .....	54
Figura 30: Rectangles contenidors dels caràcters .....	54
Figura 31: Retallat d'imatge .....	55
Figura 32: Deformació produïda pel redimensionament.....	55
Figura 33: Redimensionament equitatiu de la imatge .....	56
Figura 34: Correcció del color.....	56
Figura 35: Binarització de colors .....	57
Figura 36: Estructura bàsica d'una xarxa neuronal .....	57
Figura 37: Reducció de la mida espacial realitzada per una capa de pooling 2x2 .....	59
Figura 38: Nombres continguts en el MNIST.....	62
Figura 39: Caràcters continguts en el EMNIST.....	63
Figura 40: Aplicació de la augmentació de dades sobre una imatge .....	63
Figura 41: Comanda per executar l'eina de reconeixement d'exàmens .....	66
Figura 42: Contingut del directori d'entrada.....	67

Figura 43: Contingut del llistat d'entrada csv.....	67
Figura 44: Contingut del llistat de sortida del csv .....	68
Figura 45: Gràfica del percentatge d'encerts amb dades d'entrenament.....	69
Figura 46: Gràfica del percentatge d'encerts amb dades reals.....	70
Figura 47: Gràfica d'assignacions d'exàmens 2n Control Curs 2017-2018 Q1.....	71
Figura 48: Gràfica d'assignacions d'exàmens 3er Control Curs 2017-2018 Q1 .....	72
Figura 49: Gràfica d'assignacions d'exàmens 1er Control Curs 2017-2018 Q2 .....	73
Figura 50: Gràfica de l'evolució del nombre d'exàmens assignats.....	73
Figura 51: Gràfica temps d'entrenament requerit .....	74

## Llistat de taules

Taula 1: Estimació del temps per a cada tasca.....	28
Taula 2: Taula que conté les diferents etapes del projecte. ....	31
Taula 3: Pressupost del projecte .....	35
Taula 4: Matriu de sostenibilitat del projecte.....	36
Taula 5: Taula d'assignació de la lletra del DNI .....	65
Taula 6: Taula amb els paràmetres de l'algoritme d'associació d'exàmens .....	66

# 1 Introducció

## 1.1 Context

Aquest projecte és un Treball Final de Grau desenvolupat a la Facultat d'Informàtica de Barcelona. Aquest treball té com a principal objectiu desenvolupar una eina capaç de reconèixer el propietari d'un examen mitjançant el reconeixement visual. Aquest projecte neix de diverses necessitats identificades pel Departament d'Arquitectura de Computadors en la correcció d'exàmens.

La correcció d'exàmens a la Facultat d'Informàtica de Barcelona és una tasca pesada degut a l'elevat nombre d'alumnes que hi entren anualment. Aquest procés es podria dividir en dues fases: correcció manual per part dels professors i actualització de la puntuació dels alumnes.

- La primera etapa, correcció manual per part dels professors, és una tasca difícil d'automatitzar si no es tracta d'un examen tipus test, on sí que es podria usar el reconeixement visual sobre les caselles on s'ha de marcar la resposta correcta.
- La segona etapa, actualització de la puntuació dels alumnes, és una tasca en la qual sí que es poden aplicar diferents optimitzacions que permetrien aconseguir una reducció considerable del temps emprat. Com per exemple: l'ordenació dels exàmens segons el seu nom o la seva nota o la actualització automàtica de la seva nota final al corregir l'actual examen.

Per a poder realitzar aquestes optimitzacions anomenades anteriorment seria necessari l'ús d'una eina capaç de reconèixer de qui és l'examen que està mostrant per pantalla. D'aquesta manera al demanar a l'aplicació que ens actualitzés la nota final d'aquest alumne o que ens ordenés els exàmens per ordre alfabètic només hauria de consultar els identificadors que tingués assignats (*nomAlumne*, *nota*).

El problema és que si aquest procés d'enllaç entre la imatge del examen (examen digitalitzat) i les dades de l'alumne acaba sent més costós i feixuc que la pròpia ordenació manual dels exàmens, no val la pena la creació d'aquesta eina.

Per solucionar aquest problema és on hi prenen part les xarxes neuronals. Aquestes són capaces de realitzar el procés d'enllaç entre imatges i caràcters automàticament mitjançant l'aprenentatge reiterat.

Les xarxes neuronals com el seu nom indica intenten simular el comportament humà però amb un enfoc estratègic completament diferent. En aquest cas que el que es vol es reconèixer caràcters alfanumèrics, les xarxes neuronals es basen en recollir el major nombre d'imatges de caràcters escrits a mà per tal de posteriorment desenvolupar un sistema que pugui aprendre d'aquestes dades inicials, conegudes com a dades d'entrenament.

Un concepte important a entendre és el 'machine learning' en el qual es basa tot l'aprenentatge de les xarxes neuronals, ja que darrere d'aquest concepte hi ha tot un conjunt de tècniques

d'entrenament molt potents que fan possible que el sistema cada vegada sigui més efectiu en els seus reconeixements.

Tot i que aquest projecte neix per a satisfer les necessitats de l'assignatura d'Arquitectura de Computadors en un futur es podria exportar a totes les assignatures de la Facultat d'Informàtica de Barcelona si s'aconsegueixen uns resultats prou fiables.

Cal remarcar que en aquest projecte es pretén elaborar tota l'algorísmia necessària per a poder identificar i associar un examen al seu respectiu alumne. No obstant, aquest projecte necessita una interfície visual on l'usuari pugui corregir els exàmens de manera còmode i senzilla. Per aquest motiu durant aquest quadrimestre l'alumne de la FIB, Eric Canals, ha realitzat en paral·lel un altre Treball Final de Grau encarregat de proporcionar una eina visual de correcció d'exàmens, anomenat "*Herramienta de ayuda a la corrección de exámenes*".

El principal objectiu d'aquest projecte elaborat en paral·lel és proporcionar a l'usuari un entorn visual on poder dur a terme la correcció dels exàmens d'un mode senzill i còmode. Dintre d'aquesta eina es podran mostrar les diferents preguntes i apartats d'un examen, puntuar cadascun d'ells i afegir els comentaris oportuns.

Ambdós projectes estan molt relacionats, ja que qualsevol dels dos és indispensable per satisfer les necessitats plantejades per al professorat. Per aquest motiu durant la elaboració del Treball Final de Grau es mantindrà un feedback continu entre ambdós projectes, facilitant així la fusió d'ambdues implementacions per tal de construir una eina robusta per a la correcció d'exàmens.

## 1.2 Actors

Per a un correcte desenvolupament d'aquest projecte és necessària la implicació de diferents actors, els quals de forma més directa o indirecta aportaran informació essencial al projecte final.

### 1.2.1 Desenvolupador

És la persona més important i activa del projecte. Aquesta s'encarregarà de la recerca, la documentació i la implementació del projecte. A més a més aquesta persona també s'haurà d'encarregar de tindre una comunicació fluïda amb el codirector i director del projecte per tal d'ajustar-se als objectius esperats. En paral·lel a aquesta tasca també s'haurà de comunicar amb l'usuari final per tal d'extreure quines són les seves necessitats.

### 1.2.2 Director i codirector

Aquest dos actors són fonamentals pel correcte funcionament del projecte, ja que són les persones que coneixen els ambdós costats, és a dir, saben el que espera un tribunal d'un Treball Final de Grau i saben l'estat constantment l'estat en que es troba el projecte. Aquest últim punt i el coneixement que tenen ambdós actors en l'àmbit escollit els hi proporciona una experiència suficient com per a guiar, donar consell i ajudar al desenvolupador. En aquest

projecte les dues persones que ocuparan aquest paper seran: Josep Llosa, com a director i Carlos Álvarez, com a codirector.

### 1.2.3 Usuaris

Els usuaris principals d'aquest projecte seran les persones que acabin usant l'eina esmentada anteriorment per a fer el procés automàtic d'enllaç entre exàmens i alumnes. Aquests principalment seran els professors del Departament d'Arquitectura de Computadors, però es podria acabar extrapolant aquest ús a tots els professors de la FIB. En aquest projecte es té la possibilitat de comptar amb dos professors d'aquests departament, ja que coincideixen amb els actors d'actor i codirector.

### 1.2.4 Col·laboradors del projecte

Tal i com s'ha comentat anteriorment, aquest projecte està format per dues parts: associació automàtica d'exàmens amb el respectiu alumne i construcció d'un entorn gràfic on poder realitzar la correcció dels exàmens de forma còmode i eficient. Per aquest motiu s'ha hagut de col·laborar durant tot el quadrimestre amb l'alumne Eric Canals el qual ha estat el responsable de la construcció de l'entorn gràfic de l'eina. D'aquesta manera s'ha aconseguit que la fusió d'ambdues implementacions sigui el més senzilla possible.

### 1.2.5 Beneficiaris

Tot projecte requereix d'un beneficiari, alguna persona o grup de persones que puguin treure'n profit. En aquest cas podem distingir dos grups diferents de beneficiaris:

- Directes: En aquest cas serien els mateixos usuaris de l'eina, ja que això els hi permetrà automatitzar tasques com l'ordenació dels exàmens alfabèticament o la actualització de notes. D'aquesta manera s'aconseguirà reduir el temps dedicat aquesta tasca tant pesada.
- Indirectes: En aquest cas serien els alumnes que cursin l'assignatura d'AC (Arquitectura de Computadors) cursada a la FIB (Facultat d'Informàtica de Barcelona), ja que obtindrien les notes més ràpidament.

## 1.3 Estat de l'art

### 1.3.1 Evolució de les xarxes neuronals

Les simulacions de xarxes neuronals semblen ser un invent recent. Però el cert és que aquest és un camp que ja porta molt de temps sent explorat. Al 1943 McCulloch i Pitts van desenvolupar els primers models de xarxes neuronals basats en conceptes de neurologia mitjançant neurones artificials. Es van continuar fent estudis fins que al 1969 va arribar una etapa de frustració i desaprovació amb el llibre publicat per Minsky i Papert[1], on s'explicaven de forma generalitzada les limitacions que presentaven els models de xarxes neuronals basats en l'algoritme del perceptró que s'havien usat fins el moment.

## Perceptró

L'algoritme del perceptró va ser dissenyat per Frank Rosenblatt al 1957 continuant amb l'estudi previ de McCulloch i Pitts. Aquest semblava reproduir prou bé el comportament d'una xarxa neuronal, tot i que amb la publicació del llibre de Minsky i Papert va caure en desús.

Aquesta neurona artificial prenia com a valors diferents binaris  $x_1, x_2, \dots$ , i produïa una única sortida binària:

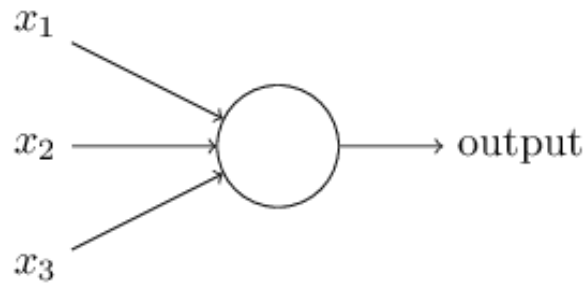


Figura 1: Model d'una neurona artificial

En aquest exemple mostrat el perceptró té 3 entrades  $x_1, x_2, x_3$ . En general poden haver-hi tantes entrades com es necessitin. Rosenblatt va proposar una regla simple per a calcular la sortida. El científic va introduir els pesos,  $w, w, \dots$ , nombres reals expressant la importància de les respectives entrades per a la sortida. La sortida de la neurona, 0 o 1, vindrà determinada per si la suma ponderada  $\sum_{j=0}^j w_j x_j$  és menor o major que el valor llindar. Aquest valor llindar també serà un nombre real propi de cada neurona. En termes algebraics:

$$\text{output} = \begin{cases} 0 & \text{if } \sum_j w_j x_j \leq \text{threshold} \\ 1 & \text{if } \sum_j w_j x_j > \text{threshold} \end{cases}$$

Durant els anys posteriors, tot i la frustració provocada pel llibre de McCulloch i Pitts es van anar fent diversos avenços importants en aquest camp que van revolucionar el món de les xarxes neuronals: neurona sigmoide i l'algoritme de backpropagation.

## Neurona sigmoide

Aquest tipus de neurona és el més usat avui en dia. Té trets molt similars al perceptró però tenen una propietat especial: els petits canvis produïts en els seus pesos o valors llindars només causen un petit canvi en la seva sortida.



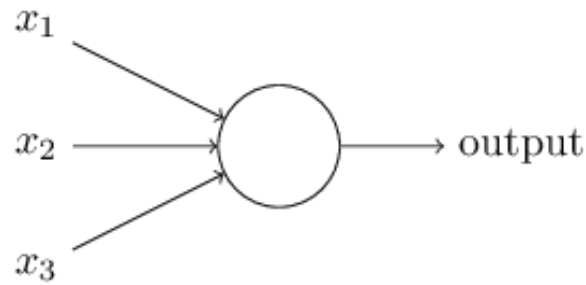


Figura 2: Model d'una neurona artificial

Les neurones sigmoïdes igual que els perceptrons, tenen entrades  $x_1, x_2, \dots$ . Però en comptes de ser únicament 0 o 1, aquestes entrades poden prendre valors reals entre 0 o 1. També com el perceptró aquesta neurona posseeix pesos per a cada entrada  $w, w, \dots$ , un valor llindar,  $b$ . Però la seva sortida no és 0 o 1, sinó que ve definida per la funció sigmoide:

$$\sigma(z) \equiv \frac{1}{1 + e^{-z}} = \frac{1}{1 + \exp(-\sum_j w_j x_j - b)}$$

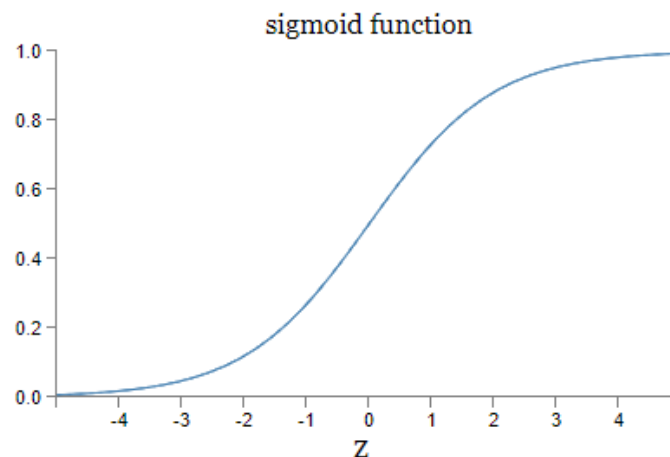


Figura 3: Gràfica de la funció sigmoide

### Backpropagation

L'algoritme de backpropagation és un algoritme que va ser introduït als 70 però no va ser apreciat fins al 1986 amb la publicació d'un document de David Rumelhart, Geoffrey Hinton i Ronald Williams. Aquest document descriu diferents xarxes neuronals on l'algoritme de backpropagation funciona molt més ràpid que en els plantejaments anteriors d'aprenentatge. Això fa possible l'ús de xarxes neuronals per a resoldre problemes que fins el moment eren impensables. Avui en dia és el mètode d'aprenentatge més usat.

Actualment les xarxes neuronals estan molt presents en el nostre dia a dia. Aquesta expansió tan sobtada en els últims anys es deu bàsicament a dos trets principals que en els anys 90 no posseïen:

- Suficients dades per a entrenar les xarxes neuronals. Fa 20 anys els jocs de dades eren 1000 vegades més petits.
- Computadors més ràpids. Avui en dia posseïm ordinadors amb molta més velocitats de càlcul que aleshores.

### 1.3.2 Eines de reconeixement de textos escrits a mà

Actualment no hi ha cap eina en el mercat que ens permeti processar els exàmens dels alumnes i fer el procés d'enllaç entre les imatges i les dades. Tot i això si que hi ha diverses eines al mercat per tal de reconèixer els caràcters escrits a mà, però el problema es que o bé no són gratuïtes o bé no són públiques.

#### ABBYY FineReader 14

[2] És un dels millors softwares de reconeixement de lletra escrita a mà del mercat. Aconsegueixen arribar a un percentatge d'encerts segons la seva pàgina web d'un 99.8%. El preu d'aquest software és de 199€.

#### Nuance's OmniPage 18

[3] Juntament amb l'anterior software esmentat són les dues empreses dominants del mercat pel que respecta al reconeixement de lletra escrita a mà. A la pàgina web no s'especifica el percentatge d'encerts. El seu preu és de 169€.

#### Recogniform Desktop Reader

[4] No és un dels software de reconeixement de lletra escrita a mà més coneguts del mercat, tot i que ofereix una gran diversitat de funcionalitats a l'hora de dur a terme tasques reconeixement visual. No s'especifica el percentatge d'encerts en la seva pàgina web. Tampoc s'especifica el preu del software.

#### Treball Final de Grau de la FIB

El quadrimestre passat es va realitzar un Treball Final de Grau a la FIB executat per l'Agustí Bau [5] que pretenia aconseguir una xarxa neuronal que proporcionés uns resultats prou acurats. Es va aconseguir una precisió del 97% en caràcters numèrics i del 91% en caràcters alfabètics. Aquests dos percentatges són insuficients per a la tasca que es vol dur a terme. Tot i que, el principal problema d'aquesta xarxa neuronal era que el disseny escollit no permetia saber els percentatges de les altres sortides en cas d'error. Això implicava que si per exemple el model proposava un DNI amb el nombre 47344499 com a sortida final i es sabia que no existia cap DNI que comencés per 4 no es podien consultar quins eren els percentatges d'encert de les altres sortides.

## 1.4 Identificació de lleis i regulacions

Els principals aspectes legals que impliquen aquest projecte són el llicenciament del codi desenvolupat, la propietat intel·lectual del mateix i la protecció de dades dels alumnes.

### 1.4.1 Llicenciament

El codi emprat en aquest projecte ha estat elaborat íntegrament. Això ens dona una gran llibertat a l'hora d'escollir la llicència del nostre projecte. Es vol apostar per una llicència de software lliure però caldrà acordar amb el professorat quina és la que s'adapta millor.

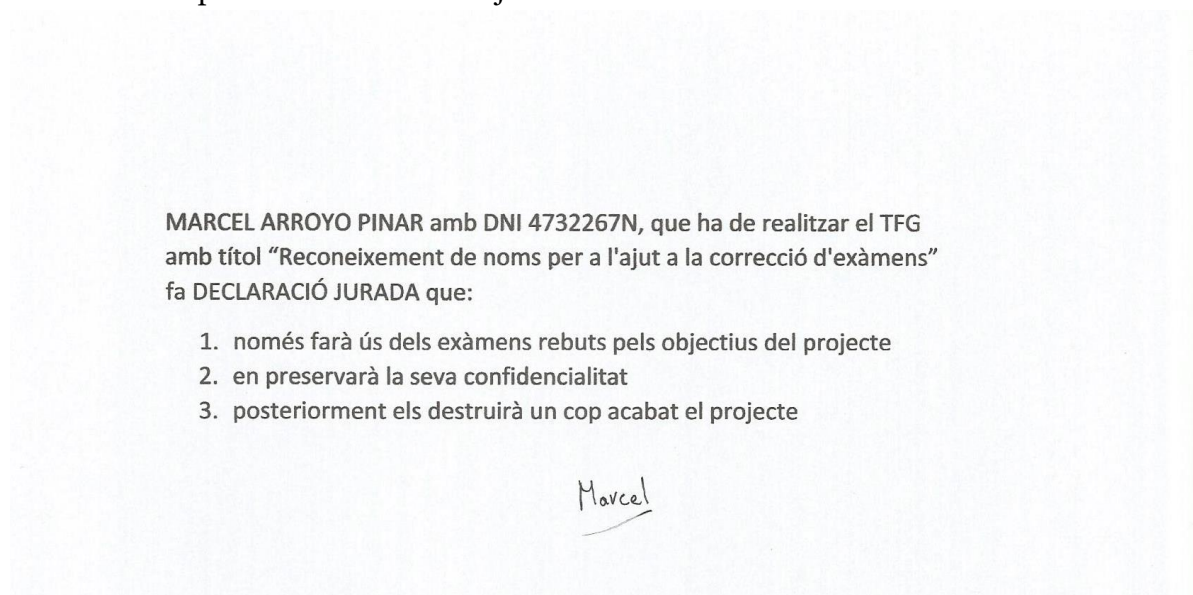
### 1.4.2 Propietat intel·lectual

La propietat intel·lectual que s'aplica sobre aquest projecte és la Normativa sobre els drets de Propietat Industrial i Intel·lectual a la UPC. Aquesta estableix un conjunt de condicions i restriccions sobre les obres dels estudiants dirigides o coordinades pel professorat de la UPC.

En aquest document s'explica quin serà el paper de la UPC respecte a la titularitat dels drets de explotació del projecte i que l'estudiant i els professors seran considerats coautors de la mateixa. També es recullen altres temes com la repartició dels beneficis derivats de l'explotació d'aquests drets.

### 1.4.3 Protecció de dades

Per al desenvolupament del projecte ha estat necessària la manipulació de dades personals. Al tractar-se de dades confidencials ha estat necessari firmar una declaració jurada per assegurar que no es farà un mal ús d'aquestes i es destruiran al finalitzar el Treball Final de Grau. A continuació es pot veure la declaració jurada:



MARCEL ARROYO PINAR amb DNI 4732267N, que ha de realitzar el TFG amb títol "Reconeixement de noms per a l'ajut a la correcció d'exàmens" fa DECLARACIÓ JURADA que:

1. només farà ús dels exàmens rebuts pels objectius del projecte
2. en preservarà la seva confidencialitat
3. posteriorment els destruirà un cop acabat el projecte

Marcel

Figura 4: Declaració jurada per a l'ús de dades personals

## 1.5 Abast del projecte

Aquest Treball Final de Grau té com a finalitat la implementació de dues xarxes neuronals amb un grau de precisió suficient per a satisfer les necessitats plantejades per el Departament d'Arquitectura de Computadors.

El percentatge d'encert que es desitja assumir en aquest projecte és un percentatge suficientment elevat per a facilitar la tasca de correcció d'exàmens. És per això que es desitja assolir un percentatge d'encerts més elevat que en antics projectes de final de carrera, on s'ha arribat al 97% d'encerts en caràcters numèrics i 91% en caràcters alfabètics en les dades d'entrenament.

Donada la limitació que presenta l'obtenció de conjunts de dades de grans dimensions per a entrenar les xarxes neuronals, es recorrerà a l'entrenament d'aquestes amb jocs de dades estàndards com són el MNIST (Modified National Institute of Standards and Technology data base)[6] i EMNIST (Extended MNIST)[7]. Ambdós models tenen caràcters molt correctes pel que fa al seu estil ortogràfic, cosa que allunya de la realitat als valors d'entrada que rebrà la xarxa neuronal al avaluar les imatges dels exàmens. El cas ideal seria l'obtenció d'un joc de dades més real, prou gran per al possible entrenament d'aquesta però degut a les limitacions de temps s'obviarà aquest punt.

Per ser conscients de quin percentatge d'encerts està dintre d'un bon llinar cal remarcar que el major percentatge aconseguit amb un model de gran complexitat a dia d'avui és del 99.79% usant conjunt de dades MNIST.

Per tal d'augmentar el percentatge final d'encerts s'utilitzaran diverses tècniques de correcció d'errors. A més, al plantejar una xarxa neuronal basada en la funció sigmoide ens permetrà un major grau de correcció al poder escollir la següent sortida amb més percentatge d'encert proposada per la xarxa neuronal.

Aquest projecte proporciona un format de sortida estàndard per facilitar el tractament d'aquestes dades en un entorn gràfic.

## 1.6 Formulació del problema

L'objectiu principal d'aquest projecte és elaborar una eina capaç d'identificar el propietari d'un examen mitjançant el reconeixement visual. Per a satisfer aquest objectiu es proporcionaran dues xarxes neuronals suficientment robustes per a la identificació de caràcters escrits a mà en un examen digitalitzat. Per a poder realitzar aquesta tasca s'haurà de fer un processat de la imatge per tal d'analitzar només els camps de l'examen que ens interessin. En aquest cas: Nom, Cognom i DNI. Una vegada identificats aquests camps en la imatge l'objectiu es reconèixer cadascun dels caràcters per a poder generar 3 identificadors (nomAlumne, cognomAlumne i dniAlumne) que corresponguin realment amb el contingut escrit a mà en aquests camps. D'aquesta manera aconseguirem associar un examen digitalitzat amb un alumne sense la necessitat de tenir una persona introduint les dades a mà d'aquests tres camps.

Aquesta associació automàtica entre l'alumne i l'examen digitalitzat aporta diverses millores en la tasca de correcció. L'ordenació automàtica dels exàmens segons el seu nom o la seva nota o l'actualització automàtica de la nota en una BD són algunes de les optimitzacions que podria aportar aquesta implementació, millorant significativament els temps de correcció dels exàmens i beneficiant tant a professors com a alumnes de la FIB.

## 1.7 Metodologia

La metodologia emprada en aquest projecte serà Scrum[8], escollida per a la seva flexibilitat a canvis inoportuns. El fet de convocar reunions cada dues setmanes amb els principals clients (director, codirector i client encarregat de la interfície) d'aquesta eina ens permeten tindre una retrospècció prou bona de com evoluciona el projecte, prendre els canvis oportuns ràpidament i complir amb les dates establertes. També cal remarcar que s'ha escollit aquesta metodologia per a poder combinar la feina laboral amb aquest projecte i així poder fixar esprints quan més convingui.

### 1.7.1 Mètodes de treball

#### Esprints de desenvolupament

Al llarg del projecte s'aniran fixant esprints en els quals en un interval de temps s'haurà enllestir la implementació establerta a priori. D'aquesta manera com s'ha esmentat anteriorment serà possible combinar l'àmbit laboral amb l'universitari. També s'aniran fixant esprints segons com es vegi l'evolució del projecte o els diferents problemes que es presentin.

#### Constant retrospècció del client

Cada dues setmanes es convoquen reunions amb el director i codirector d'aquest Treball Final de Grau, els quals alhora són els clients d'aquesta implementació. Això ens permetrà poder reorientar el projecte ràpidament i solucionar els problemes que vagin sorgint.

### 1.7.2 Eines de desenvolupament

El programa principal està desenvolupat amb Python 2.7. S'ha escollit aquesta versió, ja que a dia d'avui Python 3 continua tenint problemes de compatibilitat amb algunes de les llibreries necessàries per al tractament d'imatges. Algunes d'aquestes són:

#### Keras

Keras [9] és una API de xarxes neurals d'alt nivell, escrita en Python i capaç de funcionar amb TensorFlow, CNTK o Theano. Té tres punts forts:

- Permet prototips fàcils i ràpids (a través de la facilitat d'ús, modularitat i extensibilitat).
- Admet xarxes convolucionals i xarxes recurrents, així com combinacions de les dues.
- Funciona sense problemes amb CPU i GPU.

## OpenCV

OpenCV [10] és una llibreria open-source orientada a machine learning i visió per computadors. Conté més de 2500 algoritmes optimitzats per tal de ser usats en la detecció de rostres, classificar accions humanes en vídeos, detecció de formes...

## TensorFlow

TensorFlow [11] és una eina per a machine learning. Tot i que conté una àmplia gamma de funcionalitats, aquesta llibreria està dissenyada principalment per a models de xarxes neurals profundes.

### 1.7.3 Eines de seguiment

Per tal de fer un seguiment sobre l'evolució del projecte s'ha optat per usar dues eines molts potents pel que fa al control de versions: Git i GitHub. Aquest últim ens permet establir un repositori Git remot i un sistema de seguiment prou acceptable, ja que obliga al desenvolupador a documentar tots els canvis cada vegada que es fa un commit.

### 1.7.4 Mètode de validació

Per tal de revisar el correcte funcionament de les xarxes neuronals, una vegada entrenades amb els jocs de dades es procedirà a provar-les amb les dades de validació que inclouen ambdós jocs de proves (MNIST i EMNIST).

- En cas d'obtenir resultats prou satisfactoris, es procedirà a provar aquestes xarxes neuronals amb les imatges dels exàmens digitalitzats. Cal tenir en compte que un bon resultat amb les dades usades per a l'entrenament, no garanteix sempre un resultat final satisfactori, degut a l'ús de dades molt estandarditzades en l'etapa d'entrenament. En aquest cas es procedirà una altra vegada a l'etapa d'entrenament del model i en cas de no obtenir bons resultats s'acabarà modificant.
- En cas de no obtenir resultats prou satisfactoris es procedirà a canviar el model.

## 1.8 Integració de conceptes

Aquest projecte engloba 3 modalitats ben diferenciades: visió per computació o processat d'imatge, machine learning i algoritmes d'identificació i correcció d'errors. A continuació s'expliquen en que consisteix cadascuna d'aquestes àrees.

### 1.8.1 Processat d'imatges

El processat d'imatges consisteix en l'aplicació de diferents algoritmes amb l'objectiu de millorar la qualitat o facilitar la recerca d'informació. En aquest projecte té un pes molt important, ja que un bon processat d'imatge pot potenciar considerablement els encerts de la nostra eina de reconeixement.

### 1.8.2 Machine learning

El machine learning és una àrea que forma part de la branca de ciències de la computació. Aquesta disciplina intenta aplicar algoritmes en grans conjunts de dades per tal de descobrir patrons d'interès. Una de les tècniques emprades per aquesta disciplina és la que fa referència a les xarxes neuronals. Tal i com s'explica a l'apartat 1.3.1 les xarxes neuronals estan basades en neurones artificials que intenten simular el comportament humà. Aquest procés d'aprenentatge es realitza mitjançant l'ús de grans quantitats de dades, de les quals s'acaben inferint regles o patrons.

En aquest projecte es vol aprofitar aquesta capacitat d'aprenentatge de les xarxes neuronals per a dur a terme el reconeixement de caràcters numèrics i alfabètics escrits a mà. Per a fer-ho possible serà necessari el disseny, entrenament i avaluació de dues xarxes neuronals. Cadascuna d'aquestes estarà enfocada al reconeixement d'un tipus concrets de caràcter(dígits o números).

### 1.8.3 Correcció d'errors

La correcció d'errors en aquest projecte té una gran importància, ja que els resultats de la predicció no acostumen a ser sempre encertats. Tot i disposar d'un percentatge elevat d'encerts, hem de ser conscients que sempre hi haurà cert error en les nostres prediccions. Per aquest motiu s'han dissenyat diversos algoritmes capaços de tolerar aquest interval d'error i realitzar l'associació entre l'examen i el seu propietari.

## 1.9 Obstacles i riscos del projecte

Aquest Treball Final de Grau té com a finalitat dotar al professorat d'Arquitectura de Computadors(AC) de dues xarxes neuronals capaces de relacionar un examen digitalitzat amb l'alumne que el va realitzar. D'aquest mode s'aconseguirà solucionar el problema que té el professorat a l'hora d'ordenar i classificar els exàmens permetent-los guanyar temps per a la fase de correcció. Tot i que el problema principal a resoldre està clarament identificat hi ha altres problemes que sorgeixen a l'hora de desenvolupar una xarxa neuronal.

### 1.9.1 Dades estandarditzades

Durant l'entrenament d'ambdues xarxes neuronals s'utilitzaran conjunts de dades estandarditzats MNIST i EMNIST. Cadascun d'ells entrenarà una xarxa neuronal diferent. En el cas del MNIST entrenarà la xarxa de caràcters numèrics i en l'altre cas EMNIST entrenarà dels caràcters alfabètics. El problema d'aquest jocs de dades és que posseeixen una ortografia molt correcta, tant pel que fa a mides, com formes, mentre que les dades reals poden discernir molt d'aquests casos ideals.

### 1.9.2 Temps de computació

Les xarxes neuronals necessiten ser entrenades amb grans volums de dades. Això implica que quan més complexa sigui la nostra xarxa neuronal o bé les entrades d'aquesta, més potència de computació li haurem de proporcionar. Com en aquest projecte es compta amb uns recursos

limitats, si s'augmenta molt la complexitat dels dos factors anomenats anteriorment es pot elevar a gran escala el temps de computació per entrenar la xarxa neuronal.

### 1.9.3 Complexitat de les capes interiors

Una xarxa neuronal està formada per diverses capes. Cadascuna d'aquestes aplica una transformació al model inicial. Aquestes transformacions la majoria de vegades són operacions bastant complexes, que fa difícil saber que fa exactament cada capa. Es té una noció aproximada del que fa cadascuna d'elles però a mesura que es van concatenant es va perdent aquesta noció inicial.



## 2 Planificació temporal

En aquest apartat es descriuen les diverses tasques que es duran a terme durant els 4 mesos de projecte. També apareix una descripció temporal i una relació de dependències entre aquestes mitjançant un diagrama de Gantt. Al final d'aquesta secció, es descriuen els recursos usats per a la realització del projecte i les possibles desviacions que poden sorgir durant l'evolució d'aquest.

### 2.1 Descripció de les tasques

En aquest apartat es descriuran de forma detallada i en ordre cronològic les diferents tasques que conformen el projecte, juntament amb els recursos més essencials per a dur-les a terme. Tot i que els recursos s'explicaran de forma més detallada en el següent apartat.

#### 2.1.1 Gestió del projecte

Aquesta etapa del projecte fa referència al mòdul de GEP cursat durant l'elaboració del Treball Final de Grau. Està format per 7 entregues que permeten a l'alumne adquirir uns coneixements essencials per a tenir èxit en el seu Treball de Final de Grau. El total d'hores dedicades per aquesta tasca són 81 hores. Les entregues són les següents:

- Definició de l'abast i contextualització (26.5 hores)
- Planificació temporal (8.25 hores)
- Gestió econòmica i sostenibilitat (9.25 hores)
- Presentació preliminar (6.25 hores)
- Plec de condicions (12.5 hores)
- Presentació oral i document final (18.25 hores)

Els recursos usats per a dur a terme aquesta etapa són:

- Ordinador
- Google Docs
- Google Drive
- Adobe Acrobat Reader DC
- Càmera digital
- Gantt Project
- Racó FIB
- Atenea UPC

#### 2.1.2 Adquirir coneixements sobre xarxes neuronals

Per a la realització d'aquest projecte és necessari l'assoliment d'uns coneixements bàsics en l'àmbit de machine learning i xarxes neuronals. Durant l'inici d'aquest projecte he hagut de formar-me consultant diversos llibres en línia d'experts en aquest camp com *Neural Networks and Deep Learning*[12] i *Machine Learning Mastery*[13]. Aquests dos llibres m'han permès entendre els conceptes fonamentals de les xarxes neuronals: com estan formades, com s'entrenen elles mateixes i com avaluar la seva eficàcia. Aquest procés d'aprenentatge m'ha comportat diversos mesos, ja que treballa a jornada completa i estic cursant l'última assignatura a la FIB. Aquesta tasca només requereix d'un dispositiu a Internet per a poder consultar els dos llibres esmentats anteriorment.

#### 2.1.3 Selecció del framework de Deep Learning

Un framework de Deep Learning és una llibreria de software que ofereix diferents optimitzacions a l'hora de realitzar operacions de computació. Aquest procés de selecció, com

en l'etapa anterior, també comporta un procés d'aprenentatge autònom, ja que requereix informar-se sobre cadascun dels diferents frameworks que hi ha en el mercat i escollir-ne el més adient. Finalment en aquest projecte s'ha escollit TensorFlow, ja que s'utilitza en el reconeixement d'imatges en les que s'usen xarxes neuronals de convolucions profundes per oferir imatges amb gran precisió. Aquesta tasca requereix d'un dispositiu a Internet per a poder consultar els diferents frameworks del mercat.

#### 2.1.4 Elaboració de la xarxa neuronal

En aquesta etapa s'haurà d'elaborar una xarxa neuronal per cada tipus de caràcter: numèric i alfabètic. En primer lloc s'elaborarà la xarxa neuronal més senzilla, la qual s'encarregarà de reconèixer dígit. En segon lloc, una vegada s'hagi entrenat i avaluat la primera xarxa es procedirà a elaborar el model més complex, encarregat de reconèixer caràcters alfabètics. En aquest projecte com s'ha explicat anteriorment no és vol una xarxa neuronal que doni resposta binària sinó que es volen obtenir les diverses probabilitats dels valors de sortida donada una entrada. Segons convingui s'utilitzarà el mateix model per a la identificació de caràcters alfabètics que per a la identificació de dígit. Aquesta tasca requereix de l'ús d'un ordinador amb Python2.7 i totes les llibreries necessàries per a l'elaboració d'una xarxa neuronal. També necessita una persona formada en aquest àmbit que s'encarregui d'elaborar els diferents models.

#### 2.1.5 Entrenament de la xarxa neuronal

En aquesta etapa del projecte s'hauran d'entrenar les xarxes neuronals dissenyades anteriorment. Per tal de dur a terme aquest propòsit s'exposarà ambdues xarxes a diversos jocs de proves. En el cas de la xarxa neuronal encarregada del reconeixement de dígit, s'exposarà al conjunt de dades anomenat MNIST, el qual conté 60.000 entrades per a l'entrenament de la xarxa i 10.000 més per a l'avaluació d'aquesta. Per altra banda, la xarxa encarregada de la identificació de caràcters alfabètics, es veurà exposada al conjunt de dades anomenat EMNIST, el qual consta de 103.600 dades per a l'entrenament de la xarxa i 14.800 més per a l'avaluació d'aquesta. Aquesta tasca requereix de l'ús d'un ordinador amb Python2.7 i totes les llibreries necessàries per a l'entrenament d'una xarxa neuronal. També necessita una persona formada en aquest àmbit que s'encarregui d'elaborar els diversos entrenaments. El temps d'aquesta etapa depèn principalment del temps computacional que requereixin aquests entrenaments. Quan més complex sigui el model o les entrades d'aquest, més temps requerirà aquesta tasca.

#### 2.1.6 Avaluació de la xarxa neuronal

Aquesta tasca del projecte consisteix principalment en revisar el correcte funcionament de les xarxes neuronals. Una vegada entrenades amb els jocs de dades es procedirà a provar-les amb les dades de validació que inclouen ambdós jocs de proves (MNIST i EMNIST) per tal de veure si els resultats són els desitjats. En cas d'obtenir resultats prou satisfactoris, es procedirà a provar aquestes xarxes neuronals amb les imatges dels exàmens digitalitzats. Cal tenir en compte que un bon resultat amb les dades usades per a l'entrenament, no garanteix sempre un resultat final satisfactori, degut a l'ús de dades molt estandarditzades en l'etapa d'entrenament. En cas de no obtenir resultats prou satisfactoris es procedirà a repetir el procés d'entrenament del model. Finalment si es continua sense obtenir els resultats desitjats es modificarà el model inicial. Aquesta tasca requereix de l'ús d'un ordinador amb Python2.7 i totes les llibreries necessàries per a l'avaluació d'una xarxa neuronal. També necessita una persona formada en aquest àmbit que s'encarregui d'elaborar els mètodes d'avaluació i posteriorment prendre una decisió amb els resultats obtinguts. El temps d'aquesta etapa depèn principalment del comportament que tinguin els models davant de les dades d'avaluació.

### 2.1.7 Tractament d'errors

Aquesta tasca és una de les més importants del projecte, ja que és la que ens donarà el percentatge d'encerts final. En aquesta etapa el que es pretén és augmentar al màxim el nombre d'encerts, corregint els possibles errors que obtinguem de la xarxa neuronal. Per a fer-ho s'usaran diverses estratègies basades en tractament d'errors. Un d'aquests mètodes per a reduir el nombre d'errades serà possible gràcies al model de xarxa neuronal escollit, el qual ens permetrà obtenir les diferents probabilitats de les possibles sortides. D'aquesta manera una vegada identificat el caràcter erroni es podrà escollir la següent sortida amb més probabilitat. Aquesta tasca requereix de l'ús d'un ordinador amb Python2.7 i totes les llibreries necessàries per a la manipulació de xarxes neuronals. També necessita una persona formada en aquest àmbit que sigui capaç d'aplicar diverses tècniques de tractaments d'errors.

### 2.1.8 Etapa Final

Finalment, una vegada s'obtingui un percentatge d'encerts suficientment elevat per tal de satisfer les necessitats del Departament d'Arquitectura de Computadors es procedirà a la redacció de la memòria del projecte i la preparació de la presentació oral. Aquesta tasca requereix de l'ús d'un ordinador amb accés Internet per accedir a Google Docs i el Racó, un visualitzador de PDFs. El temps estimat són unes 70hores.

## 2.2 Estimació del temps per a cada tasca

Tasca	Estimació de temps (hores)
Gestió del projecte	81
Adquirir coneixements sobre xarxes neuronals	30
Selecció del framework de Deep Learning	10
Elaboració de la xarxa neuronal	60
Entrenament de la xarxa neuronal	60
Avaluació de la xarxa neuronal	120
Tractament d'errors	80
Etape Final	70
<b>Total</b>	<b>511</b>

Taula 1: Estimació del temps per a cada tasca

## 2.3 Diagrama de Gantt

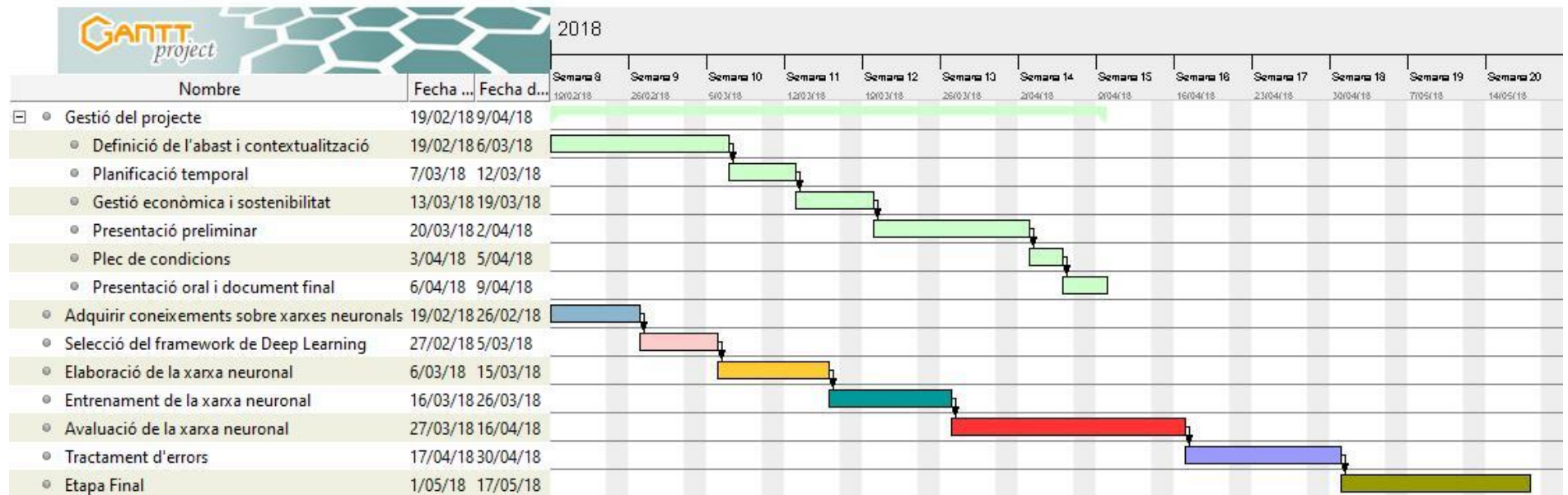


Figura 5: Diagrama de Gantt

## 2.4 Recursos

Durant el transcurs del projecte s'usaran diversos recursos que són necessaris per al desenvolupament i avaluació de les xarxes neuronals. Aquests es poden agrupar en 3 grups:

### 2.4.1 Recursos Humans

En tot projecte hi acostuma haver uns empleats encarregats de executar les idees proposades. En aquest en concret, només hi forma part un estudiant d'enginyeria informàtica especialitzat en computadors que s'encarregarà de desenvolupar els diferents models, testejar-los i fer el correcte tractament de dades per a obtenir el major percentatge d'encerts.

### 2.4.2 Recursos Hardware

Els recursos hardware usats en aquest projecte són:

- Ordinador portàtil personal: Un Acer Aspire E1-571G.
- Càmera Digital Fujifilm.
- Ordinador de sobretaula personal: Format per dos monitors LG, un ratolí, un teclat i una Workstation formada per:
  - Processador AMD FX(tm)-8150 Eight-Core 3.60Ghz
  - Memòria RAM de 16GB
  - Capacitat d'emmagatzematge de 1TB.
  - Targeta gràfica GeForce GTX 660

### 2.4.3 Recursos Software

Els recursos software usats en aquest projecte són:

- Google Docs: Editor de text per a la redacció del projecte.
- Google Drive: Gestor d'emmagatzematge per a documents.
- Adobe Acrobat Reader DC: Visualitzador de PDFs.
- GanttProject: Eina usada per a l'elaboració del digrama de Gantt
- Racó FIB: Plataforma per a la gestió del projecte.
- Atena UPC: Plataforma usada per a l'assignatura de GEP.
- Python2.7: Llenguatge de programació emprat.
- Vim: Editor de text.
- Llibreries variades: OpenCV, TensorFlow, Keras...
- Jocs de proves: MNIST(dígits) i EMNIST(lletres).

## 2.5 Modificacions respecte la planificació inicial

### 2.5.1 Desviacions

Pel que fa a la planificació temporal inicial descrita en la Taula2 ha sorgit una nova etapa que no hi apareix.

Tasca	Estimació de temps (hores)
Gestió del projecte	81
Adquirir coneixements sobre xarxes neuronals	30
Selecció del framework de Deep Learning	10
Elaboració de la xarxa neuronal	60
Entrenament de la xarxa neuronal	60
Avaluació de la xarxa neuronal	120
Tractament d'errors	80
Etape Final	70
<b>Total</b>	<b>511</b>

Taula 2: Taula que conté les diferents etapes del projecte.

La etapa s'anomena "Associació dels exàmens amb el seu respectiu alumne". Aquesta etapa es contemplava inicialment com una subetapa del Tractament d'errors, però s'ha considerat que té suficient identitat com per a esdevenir una etapa independent. Aquesta fase es situarà temporalment després del Tractament d'errors. Una vegada la xarxa neuronal presenti resultats prou satisfactoris i s'hagin aplicat les tècniques de processat d'imatge més convenients, es procedirà amb aquesta nova etapa que com el seu nom indica consistirà en relacionar cadascuna de les imatges escanejades amb un alumne del llistat. Per a poder realitzar aquesta acció s'hauran de recopilar totes les prediccions realitzades per a cada examen per arribar a construir cadascun dels camps a cercar en el llistat: DNI, nom i cognom. Una vegada enllestit aquest procediment es podrà continuar amb l'associació examen-alumne. Aquesta fase alhora englobarà l'aplicació de varies tècniques que incrementin el nombre d'encerts en aquesta associació mitjançant l'ús de diversos algorismes. Una vegada el percentatge d'encerts sigui prou satisfactori es procedirà amb la següent fase: Etapa final.

Un altre canvi que ha sorgit durant la realització del projecte és el canvi de format en la graella dels exàmens de l'assignatura Arquitectura de Computadors. Aquesta modificació s'ha realitzat per a fer més comprensible de cara a l'alumne el contingut esperat a la graella. En l'apartat X es detalla en què ha consistit aquest canvi.

### 2.5.2 Pla d'actuació

El primer canvi explicat anteriorment no té un gran impacte sobre el projecte, ja que com s'ha dit aquesta nova etapa ja estava contemplada en l'etapa de Tractament d'errors. Per fer més clara la estructura d'aquest projecte s'ha considerat que aquesta subetapa tenia suficient identitat per a esdevenir una etapa independent. Per tant pel que fa referència als recursos humans i hores dedicades del projecte no suposa cap canvi, ja que tots aquests costos es tenien ja previstos en el Tractament d'errors.

El segon canvi és que suposa una inversió de temps i recursos superior a la que es tenia prevista. Aquesta inversió recauria sobre l'etapa de tractament d'errors on s'elaboren els diferents algorismes de processat d'imatge, ja que s'haurien de reajustar diversos paràmetres

per al correcte funcionament. De totes maneres tant en la planificació com en el pressupost del projecte ja s'han considerat possibles imprevistos d'aquest tipus.

## 2.6 Pressupost

### 2.6.1 Identificació i estimació dels costos

En aquest apartat es descriuen els principals costos generats durant la elaboració del projecte. En la Taula 3, es poden veure els diferents costos agrupats per etapes del projecte, segons si són directes o indirectes i dintre d'aquests si provenen de recursos software, hardware o humans.

Costos	Unitats	Preu unitat (euros)	Vida útil (anys)	Amortització estimada (euros)**	Preu (euros)
Directes					
Gestió del projecte					
Software					
Google Docs	1	0	-	0	0
Google Drive	1	0	-	0	0
Adobe Acrobat Reader DC	1	0	-	0	0
GanttProject	1	0	-	0	0
Racó FIB	1	0	-	0	0
Atena UPC	1	0	-	0	0
Hardware					
Ordinador de sobretaula personal (dos monitors LG + ratolí + teclat + Workstation*)	1	1300	5	2.43	2.43
Ordinador portàtil personal (Acer Aspire E1-571G)	1	350	4	0.88	0.88
Càmera Digital Fujifilm	1	200	4	0.46	0.46
Recursos Humans					
Enginyer informàtic	81	35	-	-	2835
Adquirir coneixements sobre xarxes neuronals					
Software					
Adobe Acrobat Reader DC	1	0	-	0	0
Hardware					
Ordinador de sobretaula personal (dos monitors LG + ratolí + teclat + Workstation*)	1	1300	5	0.9	0.9
Ordinador portàtil personal (Acer Aspire E1-571G)	1	350	4	0.3	0.3
Recursos Humans					
Enginyer informàtic	30	35	-	-	1050
Selecció del framework de Deep Learning					



Software					
Adobe Acrobat Reader DC	1	0	-	0	0
Hardware					
Ordinador de sobretaula personal (dos monitors LG + ratolí + teclat + Workstation*)	1	1300	5	0.3	0.3
Ordinador portàtil personal (Acer Aspire E1-571G)	1	350	4	0.1	0.1
Recursos Humans					
Enginyer informàtic	10	35	-	-	350
Elaboració de la xarxa neuronal					
Software					
Python2.7	1	0	-	0	0
Vim	1	0	-	0	0
OpenCV, TensorFlow, Keras	1	0	-	0	0
Hardware					
Ordinador de sobretaula personal (dos monitors LG + ratolí + teclat + Workstation*)	1	1300	5	1.8	1.8
Ordinador portàtil personal (Acer Aspire E1-571G)	1	350	4	0.6	0.6
Recursos Humans					
Enginyer informàtic	60	35	-	-	2100
Entrenament de la xarxa neuronal					
Software					
Python2.7	1	0	-	0	0
Vim	1	0	-	0	0
OpenCV, TensorFlow, Keras	1	0	-	0	0
MNIST(dígit) i EMNIST(lletres)	1	0	-	0	0
Hardware					
Ordinador de sobretaula personal (dos monitors LG + ratolí + teclat + Workstation*)	1	1300	5	1.8	1.8
Ordinador portàtil personal (Acer Aspire E1-571G)	1	350	4	0.6	0.6
Recursos Humans					
Enginyer informàtic	60	35	-	-	2100
Avaluació de la xarxa neuronal					
Software					
Python2.7	1	0	-	0	0
Vim	1	0	-	0	0
OpenCV, TensorFlow, Keras	1	0	-	0	0

MNIST(dígitos)	i	1	0	-	0	0
EMNIST(lletres)						
Hardware						
Ordinador de sobretaula personal (dos monitors LG + ratolí + teclat + Workstation*)		1	1300	5	3.6	3.6
Ordinador portàtil personal (Acer Aspire E1-571G)		1	350	4	1.2	1.2
Recursos Humans						
Enginyer informàtic		120	35	-	-	4200
Tractament d'errors						
Software						
Python2.7		1	0	-	0	0
Vim		1	0	-	0	0
OpenCV, TensorFlow, Keras		1	0	-	0	0
MNIST(dígitos)	i	1	0	-	0	0
EMNIST(lletres)						
Hardware						
Ordinador de sobretaula personal (dos monitors LG + ratolí + teclat + Workstation*)		1	1300	5	2.40	2.40
Ordinador portàtil personal (Acer Aspire E1-571G)		1	350	4	0.81	0.81
Recursos Humans						
Enginyer informàtic		80	35	-	-	2800
Etapla Final						
Software						
Google Docs		1	0	-	0	0
Google Drive		1	0	-	0	0
Adobe Acrobat Reader DC		1	0	-	0	0
Racó FIB		1	0	-	0	0
Atena UPC		1	0	-	0	0
Hardware						
Ordinador de sobretaula personal (dos monitors LG + ratolí + teclat + Workstation*)		1	1300	5	2.1	2.1
Ordinador portàtil personal (Acer Aspire E1-571G)		1	3502	4	0.7	0.7
Recursos Humans						
Enginyer informàtic		70	35	-	-	2450
Indirectes						
Llum		4	8	-	-	32
ADSL		4	60	-	-	240
Total acumulat						18117.98

Contingència		5% Total acumulat			905.89
Imprevistos***	2	1050	10% risc	-	210
Total sense IVA					19233.87
Total amb IVA(21%)					23272.98

Taula 3: Pressupost del projecte

(\*): Les característiques de la Workstation estan descrites en l'apartat 1.4.2. Recursos Hardware de la entrega anterior.

(\*\*): La amortització estimada està avaluada respecte les hores aproximades descrites a l'apartat 1.2 Estimació del temps per a cada tasca de la entrega anterior.

(\*\*\*): Els imprevistos que apareixen a la Taula 3 fan referencia a les etapes: Avaluació de la xarxa neuronal i Tractament d'errors.

### 2.6.2 Control de gestió

Com s'ha vist en apartats anteriors, el nostre projecte pot patir modificacions imprevistes, les quals també es veuen reflectides en moltes ocasions en el nostre pressupost. Per això és important prevenir aquests fets inesperats amb mecanismes que ens permetin reconduir el nostre projecte a l'èxit.

En el nostre cas, ens hem centrat en les etapes que són més propenses a tenir un desviació inesperada: Avaluació de la xarxa neuronal i Tractament d'errors. Per ambdues s'ha pressupostat un imprevist amb un percentatge del 10% tal i com es pot apreciar a la Taula 3. El motiu d'aquests imprevistos són deguts a que ambdues tasques són les més crítiques del projecte i per tant en les que poden sorgir més problemes.

Respecte el percentatge de contingència emprat, s'ha considerat que un 5% és un indicador suficient degut al detall emprat a l'hora de identificar i estimar els diferents costos per a cadascuna de les tasques del projecte.

## 2.7 Sostenibilitat i compromís social

La sostenibilitat i compromís social són dos factors importants a considerar en qualsevol projecte. Per aquest motiu s'ha analitzat quin impacte té aquest projecte en el nostre entorn. Aquest estudi s'ha realitzat basant-se en tres dimensions: econòmica, ambiental i social. En cadascuna d'aquestes s'ha obtingut una puntuació tal i com es pot veure a la Taula 4.

Dimensió	PPP	Desmantellament	Riscos
Econòmica	Pressupost	Pla viabilitat	Riscos econòmics
	8.5	18	0
Ambiental	Consum energètic	Petjada ambiental	Riscos ambientals
	8	15	0
Social	Impacte personal	Impacte social	Riscos socials
	8	15	0
Rang de sostenibilitat	<b>24.5</b>	<b>48</b>	<b>0</b>
	<b>72.5</b>		

Taula 4: Matriu de sostenibilitat del projecte

### 2.7.1 Dimensió econòmica

En aquest document es recull un pressupost prou acurat que descriu els diferents costos del nostre projecte. Aquesta avaluació, visible en l'apartat de Identificació i estimació dels costos té en compte els diferents recursos emprats i els imprevistos que poden sorgir. El pressupost establert no és ambiciós més aviat està molt ajustat a les necessitats bàsiques per a fer possible aquest projecte.

En termes de competitivitat, no és un projecte que busqui entrar al mercat, ja que està pensat per satisfer unes necessitats concretes del professorat del Departament d'Arquitectura de Computadors. Tot i que en un futur la resta de professorat de la FIB podria usar aquesta eina.

Actualment hi ha algunes llibreries OCR al mercat que són capaces de reconèixer caràcters escrits a mà com s'ha mencionat a l'apartat 1.3.2, tot i que la majoria d'elles són de pagament. La principal millora econòmica que planteja aquest projecte és una significativa reducció del temps emprat durant la correcció d'exàmens, podent emprar aquest temps per a realitzar altres tasques productives.

### 2.7.2 Dimensió ambiental

L'únic recurs que consumeix aquest projecte és la llum. Depenent de l'origen de l'energia elèctrica emprada aquest impacte pot ser més o menys rellevant. També es poden considerar els materials emprats en la elaboració dels components electrònics d'ambdós ordinadors els quals normalment són productes que una vegada exhaurit el temps de vida útil és complex el seu desmantellament.

Tot i que els aparells electrònics sí que s'han inclòs en el pressupost, no ha calgut la compra de cap d'ells, ja si s'han emprat aquests és justament pel fet que eren recursos que es podien aprofitar per a realitzar aquest projecte.

Aquest projecte es podria reciclar per a qualsevol altre projecte encarregat del reconeixement de caràcters, però en l'àmbit ambiental aquesta proposta no té cap impacte significatiu.

### 2.7.3 Dimensió social

Aquest projecte m'ha permès descobrir un nou món que fins el moment no havia explorat: machine learning mitjançant xarxes neuronals. És un sector que a dia d'avui està molt present en les nostres vides i està en constant expansió. Per tant, ha estat una bona decisió endinsar-se en aquesta branca de la informàtica.

Aquest projecte té la intenció de canviar la metodologia de correcció del professorat de l'assignatura d'Arquitectura de Computadors, fent-la més senzilla i hàbil. D'aquesta manera s'aconseguirà millorar la qualitat de vida tant per a professors com per a alumnes.

A nivell global pot ser que aquest projecte no canviï absolutament res, no obstant es pot donar el cas en que aquest mètode de correcció d'exàmens acabi sent la metodologia emprada en quasi tots els departaments de la FIB.

## 3 Disseny de l'eina

### 3.1 Integració del desenvolupament en el context de l'aplicació

Tal i com s'ha explicat a l'apartat 1.1 el projecte està format per dos Treballs Finals de Grau, un encarregat d'elaborar l'algorísmia d'identificació d'exàmens i l'altre per a l'elaboració d'un entorn gràfic on poder gestionar i corregir els exàmens de forma senzilla i còmode.

Per aquest motiu ha estat necessari desenvolupar aquest projecte sota unes condicions que facilitin la incorporació d'aquesta algorísmia en el entorn gràfic dissenyat paral·lelament per l'Eric Canals. A més també s'ha hagut de tenir en compte l'entrada proporcionada pel professorat de manera que el procés de lectura no impliqui una gran càrrega de treball per aquests durant l'execució d'aquesta eina.

El professorat de l'assignatura d'Arquitectura de Computadors genera un parell o tres de documents digitalitzats en format PDF, els quals contenen tots els exàmens dels alumnes que s'han presentat a l'avaluació. Aquesta tasca no implica cap carrega de treball per al professorat ja que tot i que el procés d'escanejat sigui lent, és automàtic i es poden realitzar altres tasques paral·lelament. Aquests documents digitalitzats són separats en diferents arxius PDF, un per a cada examen. Aquest procés és automàtic i està realitzat per l'eina generada per l'Eric Canals. L'última tasca que ha de realitzar el professorat és generar un fitxer csv on hi figurin les dades dels alumnes matriculats a l'assignatura. Aquest procés tampoc requereix d'una gran quantitat de temps, ja que la generació d'aquest arxiu és pot realitzar de forma automàtica amb les dades dels alumnes.

L'eina elaborada pel company, realitza una crida al programari encarregat de la identificació d'exàmens, el qual s'encarrega de llegir el llistat csv d'entrada i cadascun dels arxius PDF generats. Una vegada realitzada la identificació d'exàmens es genera un arxiu de sortida. Aquest arxiu té el mateix format que el d'entrada però amb un camp extra on hi figura el nom de l'arxiu que conté l'examen associat a aquell alumne. El format d'aquest arxiu també és un llistat csv per tal de facilitar la lectura de dades a l'entorn gràfic generat per l'Eric Canals.

Una vegada generat el llistat de sortida on hi figuren les diferents associacions d'exàmens realitzades, és on comença la lectura i processament d'aquestes dades per a la construcció d'un entorn gràfic on poder realitzar el procés de correcció d'exàmens més senzill i còmode.

### 3.2 Criteris de disseny

Durant tot el projecte s'han hagut d'anar prenent diferents decisions de disseny que han afectat directament a l'eina de reconeixement d'exàmens. Aquestes decisions s'han agrupat segons el camp on han hagut d'estar preses.

### 3.2.1 Processat d'imatge

#### OpenCV

En aquest projecte s'ha optat per treballar amb OpenCV ja que presenta un rendiment més elevat que Matlab. Degut a que hem de processar una quantitat considerable d'imatges s'ha considerat oportú emprar OpenCV per la seva eficiència.

Aquesta llibreria ens permetrà realitzar tot tipus de transformacions en les nostres imatges: binaritzacions, cerca de formes en la imatge, redimensionaments...

#### Wand

Els exàmens proporcionats per al professorat de l'assignatura d'Arquitectura de Computadors són generats en format PDF degut a la digitalització de la impressora. Aquesta genera diversos arxius PDFs, que contenen tots els exàmens dels alumnes.

OpenCV no és capaç de tractar arxius PDF degut a la seva complexitat. Per aquest motiu s'ha emprat la llibreria Wand [14] la qual ens permet tractar diferents formats d'imatge i convertir-ho al format desitjat. En aquest projecte s'ha considerat treballar amb imatges JPEG (Short for Joint Photographic Experts Group) degut al poc espai d'emmagatzematge requerit.

#### Copy

Les instruccions d'assignació de Python no copien objectes, creen enllaços entre un objectiu i un objecte. Per a conjunts que són mutables o que contenen elements mutables, de vegades es necessita una còpia, de manera que es pot modificar l'objecte original sense canviar l'altre. Per aquest motiu s'ha emprat aquest mòdul el qual proporciona operacions genèriques de còpia profunda.

#### Os i sys

Durant aquesta fase del projecte és necessari realitzar lectura d'imatges al nostre programa. Els mòduls os i sys proporcionen nombroses eines per tractar fitxers, rutes i directoris. El mòdul os conté dos submòduls os.sys (igual que sys) i os.path que es dediquen al sistema i als directoris respectivament.

### 3.2.2 Machine learning

#### Xarxes neuronals convolucionals

Les xarxes neuronals convolucionals acostumen a estar compostes per una o més capes convolucionals. La arquitectura que segueixen aquestes xarxes està dissenyada per aprofitar la estructura 2D d'una imatge d'entrada. A més acostumen a ser fàcils d'entrenar i tenen molts menys paràmetres que altres xarxes neuronals amb el mateix nombre de capes intermèdies. Una altra avantatge d'aquests models és que les imatges emprades per aquestes xarxes no necessiten tenir un processament massa acurat.

Com el nostre projecte s'encarrega únicament de reconèixer imatges 2D s'ha optat per adoptar aquesta arquitectura a la nostra xarxa neuronal.

### Capa d'activació sigmoide

Un dels propòsits d'aquest projecte és proporcionar les probabilitats de totes les opcions d'una predicció, és a dir, que al realitzar una predicció no només obtinguem la opció correcta, sinó totes les possibilitats. En altres projectes s'usa la capa d'activació softmax, la qual amplia la diferència entre la probabilitat més elevada i la resta. En aquest projecte el que es pretén és obtenir aquestes probabilitats per a que en cas d'error es pugui consultar quin seria el següent candidat i modificar la predicció. Per aquest motiu s'emprarà la capa d'activació sigmoide la qual és molt emprada per descriure casos d'aprenentatge de sistemes complexos que mostren una progressió temporal.

### TensorFlow

TensorFlow és una biblioteca de software de codi obert per al càlcul numèric que utilitza grafs de flux de dades. Els nodes en el graf representen operacions matemàtiques, mentre que les arestes del graf representen conjunts de dades multidimensionals (tensors) comunicats entre ells. L'arquitectura flexible li permet realitzar càlculs en una o més d'una CPU o GPU d'un ordinador, servidor o dispositiu mòbil des de la mateixa API. TensorFlow va ser desenvolupat originalment per investigadors i enginyers que treballaven a l'equip Brain de Google destinat a la realització d'investigacions d'aprenentatge autònom i xarxes neuronals profundes.

A la pàgina web de TensorFlow hi ha força documentació sobre xarxes neuronals convolucionals, es poden trobar exemples d'implementacions, tutorials i descripcions força acurades sobre les diferents funcions de la llibreria.

Actualment hi ha més de 6000 repositoris de codi obert que usen aquesta eina. Per aquest motiu i per l'enorme quantitat d'informació que hi ha sobre TensorFlow s'ha decidit emprar aquest framework pel nostre projecte.

### Keras

Keras és una API de xarxes neuronals escrita en Python. És un wrapper capaç d'executar-se sobre TensorFlow, CNTK, o Theano. Concretament es farà servir en conjunt amb Tensorflow. Suporta les xarxes neuronals convolucionals i les xarxes recurrents, a més de permetre l'execució tant en CPU com en GPU. Els objectius que persegueix el framework de Keras són principalment: ser una plataforma user-friendly i de fàcil extensibilitat. A més, té un gran compatibilitat amb Python, ja que suporta les versions 2.7 i 3.6.

Degut a la seva gran compatibilitat amb Python i Tensorflow, s'ha acabat emprant el framework Keras.



## Numpy

Durant aquest projecte s'hauran de tractar constantment estructures de N-dimensionalitat. Per aquest motiu s'ha decidit emprat la llibreria NumPy. Aquesta és fonamental per a la informàtica científica amb Python. Conté entre altres coses:

- un potent tractament d'objectes d'N-dimensions
- funcions sofisticades
- eines per a la integració de codi C/C++ i Fortran
- àlgebra lineal útil, transformació de Fourier i capacitats de nombres aleatoris

### 3.2.3 Correcció d'errors

#### Interval d'errors en les prediccions

Per tal de poder mesurar l'error que hi ha entre una predicció i un valor real, es va adoptar un sistema ponderat de distàncies de fallada. Aquest algoritme consisteix bàsicament en calcular quants canvis s'haurien de realitzar per a que la predicció acabes sent el valor real. Aquests canvis poden consistir en un intercanvi d'una lletra, la supressió o inserció d'una lletra. Cadascuna d'aquestes modificacions té una ponderació, de mode que una vegada executat l'algoritme obtenim un valor que cal comparar amb un threshold per saber si s'accepta la predicció inicial.

#### Fiabilitat

Per tal de que aquesta eina sigui un producte usable i eficient, es vol aconseguir un marge de fiabilitat, és a dir, intentar que no hi hagi cap examen que s'assigni incorrectament. Per aquest motiu s'han emprat uns thresholds i unes ponderacions en els algoritmes de correcció bastant conservadors, ja que una assignació incorrecta implica un procés feixuc per a l'usuari.

#### Csv

Per tal de realitzar la correcció d'errors serà necessari disposar dels valors reals amb els que poder comparar les prediccions realitzades. El professorat de l'assignatura d'Arquitectura de Computadors, ens ha proporcionat el llistat csv (comma-separated values) que es genera automàticament amb les dades dels alumnes matriculats. Per tal de poder llegir aquest document fàcilment s'usarà la llibreria csv, el qual implementa classes per llegir i escriure dades tabulars en format csv.

## 3.3 Disseny de l'eina

En aquest apartat es pretén explicar quines són les principals etapes que realitza la nostra aplicació i una breu descripció de cadascuna d'elles.

### 3.3.1 Lectura

Aquest procés consisteix en llegir totes les entrades necessàries per a que l'aplicació pugui realitzar la seva funció. Aquest procés requereix la lectura d'un arxiu csv on hi ha de figurar les dades dels alumnes (nom, cognom i dni) i els documents digitalitzats corresponents als exàmens dels alumnes. En cas de no complir-se algun d'aquests requisits es suspendrà l'execució del programa. D'altre manera es generarà un arxiu JPG per a cada primera plana dels documents digitalitzats.

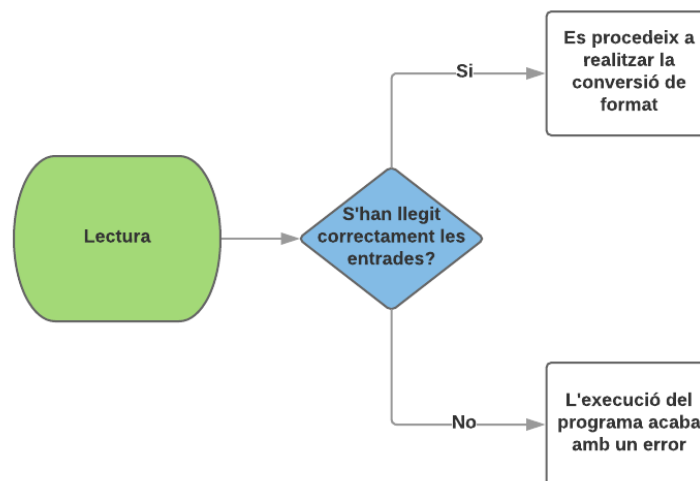


Figura 6: Diagrama de flux corresponent a l'etapa de lectura

### 3.3.2 Identificació de la graella

Aquest procés consisteix en extreure les coordenades corresponents al contorn de la graella on hi figuren les dades personals dels alumnes (nom, cognom i dni). Per a dur a terme aquesta tasca s'ha retallat la imatge inicial per la zona aproximada on hi figura la graella. S'ha aplicat una binarització de color i una suavització de la imatge per tal d'extreure la informació rellevant i eliminar possibles impureses de la imatge. Seguidament s'ha procedit a identificar les línies horitzontals i verticals de llarga longitud de la imatge eliminant la resta de components, per tal de reconstruir el contorn de la graella. Una vegada només hi figura el contorn de la graella a la imatge s'han obtingut les coordenades mitjançant una cerca.

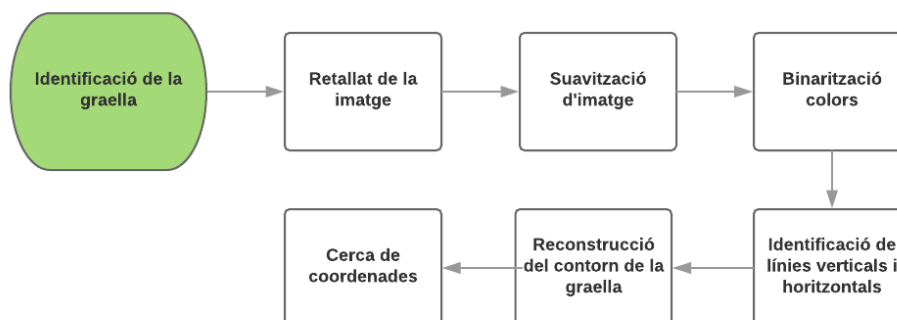


Figura 7: Diagrama de flux de la identificació de la graella

### 3.3.3 Identificació dels camps i les caselles

Aquest procés consisteix en identificar els diferents camps de la graella, i posteriorment dividir cadascun d'aquests entre el nombre corresponent de caselles que continguin. D'aquesta manera aconseguirem la casella contenidora de la lletra a predir.

### 3.3.4 Tractament de la imatge

Una vegada obtinguda la casella contenidora de la imatge a predir cal aplicar diversos tractaments a aquesta imatge per a fer-la lo més semblant possible a les imatges que s'han usat per a entrenar la xarxa neuronal amb la que es predirà quin caràcter conté la imatge. Alguns d'aquest procediments són:

- Retallat d'imatge: Consistirà en retallar els marges de la imatge per a no mostrar les línies de la graella
- Redimensionament equitatiu: s'afegiran píxels als marges fins aconseguir una imatge quadrada de mode que al fer un redimensionament no es produeixi cap deformació.
- Correcció de color: Es considerarà el valor màxim de la imatge com a el valor màxim de referència, en el cas de les imatges 255. A partir d'aquest valor mitjançant una regla de 3 es calcularà la resta de colors.
- Binarització de colors: Consistirà en convertir tots els píxels de la imatge en el valor màxim (blanc) o mínim (negre) a partir d'un threshold, per a fer les imatges més similars a les que s'usen per entrenar la xarxa neuronal estan molt contrastades.
- Identificació de caselles buides: Detectar si es tracta d'una casella buida o no. En cas de que ho sigui no es predirà el contingut de la casella.

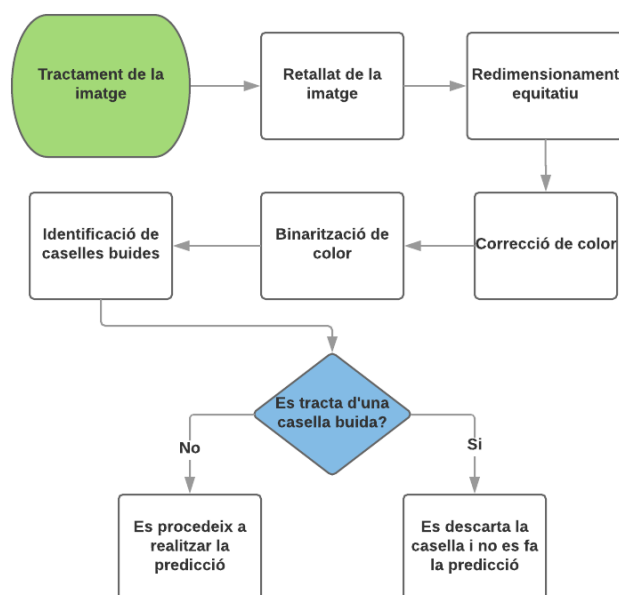


Figura 8: Diagrama de flux del tractament de la imatge

### 3.3.5 Entrenament de la xarxa neuronal

Tot i que aquest procés no es realitza en la pròpia eina d'identificació, per tal de predir el contingut d'una imatge s'ha hagut d'entrenar una xarxa neuronal anteriorment. Aquest

procediment consisteix en entrenar la xarxa neuronal reiteradament amb jocs de dades diferents, fins a obtenir uns percentatges d'encerts satisfactoris. Com només disposem del joc de proves MNIST per a caràcters numèrics i del EMNIST per a caràcters alfabètics s'han hagut de modificar aquests jocs de proves mitjançant l'augmentació de dades.

L'entrenament al que han estat sotmesos els diferents models ha estat sotmès al següents paràmetres:

- Epoch: Nombre d'iteracions que s'entrenarà el model amb les dades d'entrenament. En aquest projecte s'ha seleccionat 5 iteracions.
- Batch size: Nombre d'exemples d'entrenament que s'estan provant a l'hora. Quant més gran sigui la mida d'aquest conjunt més memòria consumirà el procés d'entrenament.
- Steps per epoch: Nombre d'iteracions per a cada epoch. En aquest mètode d'entrenament es realitzen 937 iteracions. Aquest nombre sorgeix de dividir el total de dades d'entrenament entre la mida del batch.

En aquest projecte s'ha utilitzat una funcionalitat de Keras que únicament guarda el model entrenat quan els percentatges d'encert superen els aconseguits anteriorment. Aquests models es guarden mitjançant arxius amb el format h5.

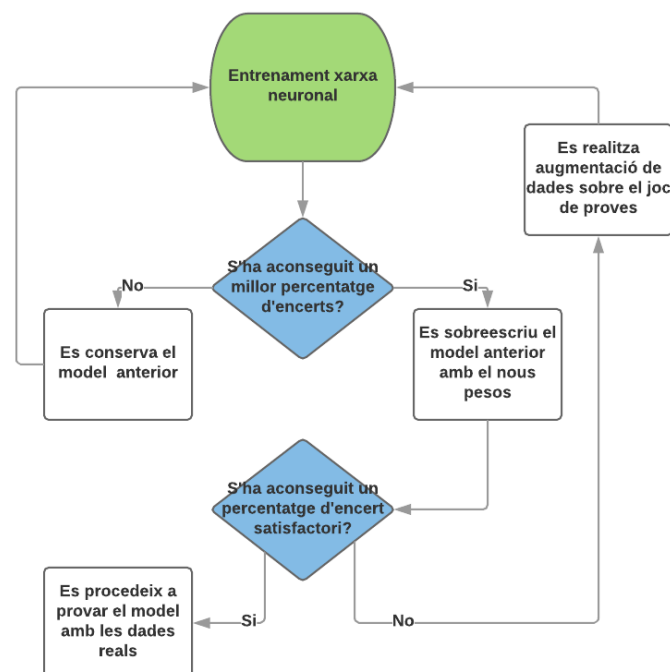


Figura 9: Diagrama de flux de l'entrenament de la xarxa neuronal

### 3.3.6 Predicció de caràcters

Aquesta etapa consisteix en la identificació de caràcters escrits a mà en una imatge mitjançant la xarxa neuronal elaborada i entrenada anteriorment. La diferència principal amb la fase d'entrenament és que en aquesta etapa les dades són reals, és a dir, les imatges provenen d'un document digitalitzat i no pas d'un joc de dades ideat per a l'entrenament de xarxes neuronals.

En aquesta fase en base al tipus d'imatge a predir (caràcter alfabètic o numèric) es selecciona la xarxa adient per a cada tipus de caràcter i es procedeix amb la seva predicció.

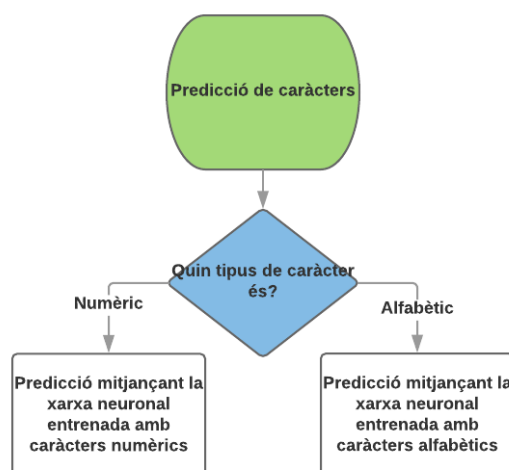


Figura 10: Diagrama de flux de la predicció de caràcters

### 3.3.7 Reconstrucció de les dades personals mitjançant les prediccions

Una vegada obtingudes les prediccions de totes les caselles contenidores de caràcters cal reconstruir el nom, cognom i DNI de l'alumne per tal de poder comparar els resultats amb els alumnes matriculats a l'assignatura que figuren en el llistat csv citat anteriorment.

### 3.3.8 Correcció d'errors i assignació d'exàmens

Aquesta etapa consistirà en assignar els documents digitalitzats (exàmens) amb algun dels alumnes que figuren en el llistat csv. Aquesta assignació es realitzarà en base als resultats obtinguts. Com la nostra xarxa neuronal no presenta un encert del 100%, les prediccions obtingudes presentaran un cert error. Per aquest motiu s'han elaborat diversos algoritmes que realitzen aquesta assignació d'exàmens permetent cert marge d'error.

Per tal d'assignar un examen a una predicció es seguiran aquests passos:

1. Es calcularà la lletra generada mitjançant els nombres del DNI predit. La lletra del DNI espanyol es generada mitjançant un algoritme que usa la resta de nombres del document d'identificació, ja que és un dígit de control. En cas de que la lletra predita coincideixi amb la lletra calculada es procedirà a buscar el DNI al llistat csv. En cas de que hi figuri al llistat, s'assignarà el document digitalitzat a l'alumne que tingui el mateix document d'identificació.

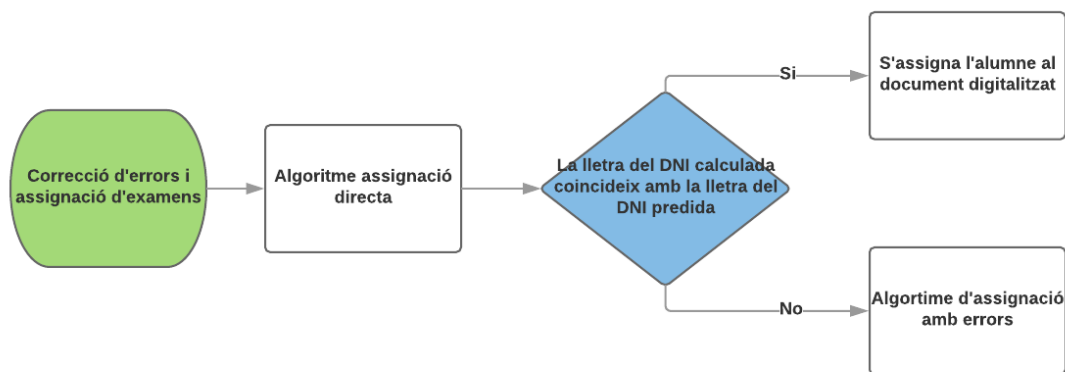


Figura 11: Diagrama de flux de l'algoritme d'assignació directa

2. En cas de que es trobi al llistat un DNI que difereixi en un únic caràcter amb el DNI predit, aquesta alumne passa a ser possible candidat per a l'assignació. Si al recórrer tots els alumnes només s'ha trobat un únic candidat el document digitalitzat es assignat a aquell alumne.

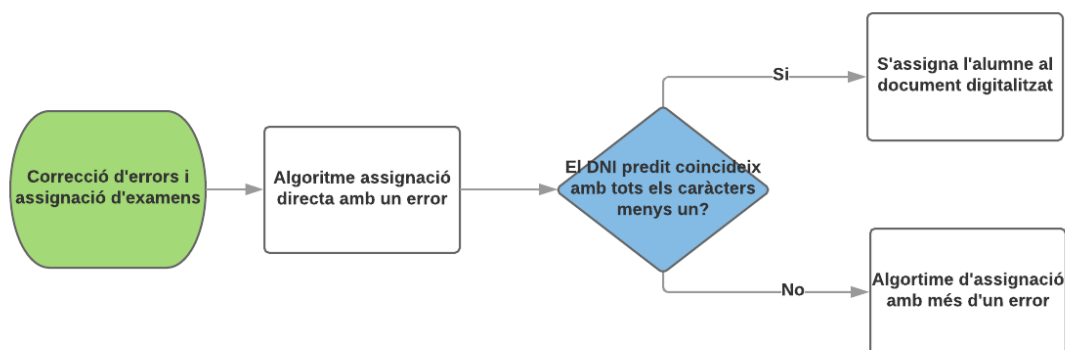


Figura 12: Diagrama de flux de l'algoritme d'assignació directa amb un error

3. En cas de que el DNI predit presenti 2 o més errors amb qualsevol DNI del llistat es sotmetrà a l'últim algoritme d'identificació proposat en aquest projecte. Aquest algoritme consisteix en ponderar els diferents canvis que s'haurien d'aplicar a la predicció del nom i cognom de l'alumne obtingudes, per tal d'acabar corresponent a algun dels alumnes que encara no han estat assignats. Amb aquestes ponderacions es calculen 2 distàncies: nom i cognom, les quals vindrien a descriure la quantitat d'error que hi figura en la predicció. Si aquestes distàncies són inferiors als thresholds proposats s'assigna el document digitalitzat a l'alumne. Aquest algoritme s'executa de forma seqüencial mentre s'augmenta el nombre d'errors permesos. D'aquesta manera aconseguim un filtratge més eficaç, ja que a cada iteració es redueix el nombre de possible candidats.

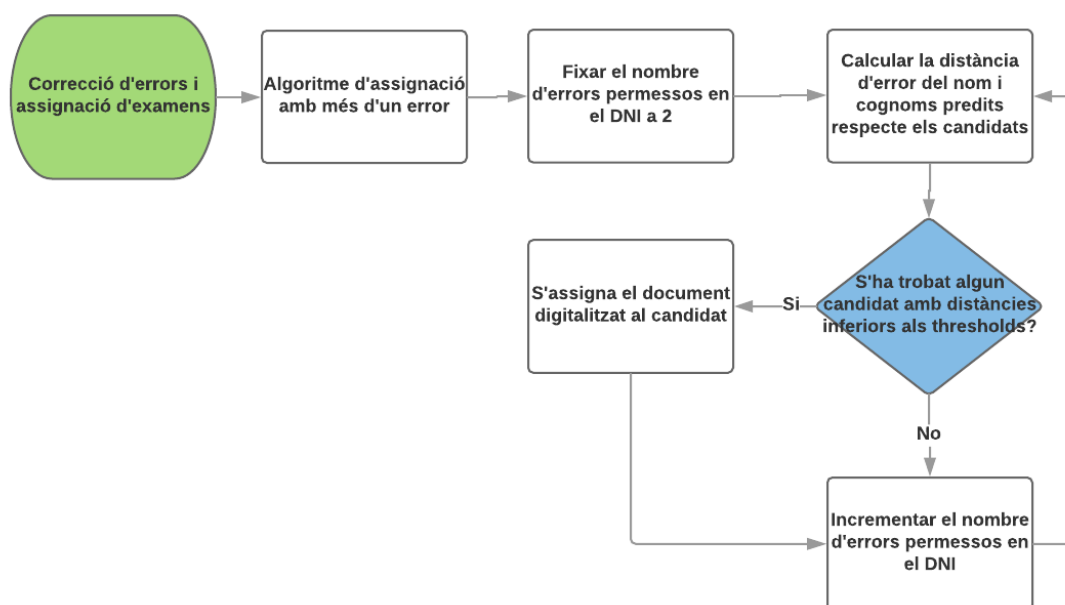


Figura 13: Diagrama de flux de l'algoritme d'assignació amb més d'un error

### 3.3.9 Sortida

Aquesta última etapa consisteix en la generació dels resultats respecte les assignacions establertes durant l'etapa anterior. Per tal de facilitar la lectura d'aquests resultats a possibles aplicacions futures, com l'eina elaborada per l'Eric Canals, s'ha decidit guardar les dades en un llistat csv. Aquest arxiu seguirà el mateix format que el llistat csv d'entrada però comptarà amb un camp extra on hi figurarà el nom del arxiu corresponent al examen assignat. En cas de no haver trobat cap examen hi figurarà la paraula "None".

## 4 Implementació

En aquest apartat es pretén explicar els aspectes més rellevants respecte la implementació de l'eina proposada en aquest projecte. Els diferents apartats es corresponen amb les tasques realitzades per a assolir l'objectiu principal d'aquest projecte.

### 4.1 Digitalització d'imatges

Aquesta etapa consisteix en la digitalització dels exàmens dels quals es vol dur a terme el procés d'identificació de caràcters. Tot i que és una etapa que està implícita cal tenir en compte que el processat d'imatges que es contempla en aquest projecte assumeix que les imatges digitalitzades tindran unes característiques similars i es respectarà un cert llindar de qualitat de la imatge. Aquest procés ha estat realitzat pel professorat d'Arquitectura de Computadors mitjançant una de les impressores situada a l'edifici D6.

Els documents digitalitzats contenen els exàmens corresponents a:

- 2n Control Arquitectura de Computadors Curs 2017-2018 Q1
- 3er Control Arquitectura de Computadors Curs 2017-2018 Q1
- 1er Control Arquitectura de Computadors Curs 2017-2018 Q2

Cadascun dels controls digitalitzats ha estat proporcionat mitjançant tres PDFs, cadascun dels quals aproximadament conté 60 exàmens, excepte l'examen corresponent al Q2 que s'han entregat 4 PDFs degut al gran nombre d'alumnes matriculats. A continuació es mostra un dels exàmens corresponent al control del Q2 (Figura 14) i un del Q1 (Figura 15).

2n Control Arquitectura de Computadors Curs 2017-2018 Q1

COGNOM:

NOM:  DNI:

Problema 3, (4 punts)

Tenemos un procesador (que llamaremos Pmp) sin memoria cache, todos los accesos se realizan directamente sobre memoria principal. En este procesador hemos ejecutado una aplicación A (que usaremos a lo largo de todo el problema) y hemos obtenido los siguientes datos: tiempo de ejecución  $18 \times 10^9$  instrucciones ejecutadas,  $10^{10}$  accesos a memoria (incluye tanto los accesos a datos como los accesos a instrucciones). Sabemos además que Pmp funciona a una frecuencia de 2 GHz. Para simplificar el problema supondremos que todos los accesos son lecturas.

a) Calcule el CPI del procesador Pmp al ejecutar la aplicación A (CPI<sub>app</sub>).

b) Calcule el tiempo de penalización de los accesos a memoria (en ciclos) del procesador Pmp respecto al Pideal.

c) Calcule el tiempo de ejecución de la aplicación A en el procesador P2a.

Figura 14: Format d'examen corresponent al 2n Control del Curs 2017-2018 Q1



1er Control Arquitectura de Computadors Curs 2017-2018 Q2

COGNOMS:

NOM:  DNI/NIE:

**IMPORTANT** Leer atentamente antes de empezar el examen. Escribe los apellidos, el nombre y el DNI/NIE antes de empezar el examen. Escribe un solo carácter por recuadro, en mayúsculas y lo más claramente posible. Es importante que no haya tachones ni borrones y que cada carácter quede remarcado dentro de su recuadro sin llegar a tocar los bordes. Usa un lápiz cuado en blanco para separar los apellidos y nombres componentes si es el caso. No escribas fuera de los recuadros, todo lo que haya fuera de ellos es ignorado. La identificación del alumno se realiza de forma automática, no seguir correctamente estas instrucciones puede comportar no tener nota.

**Problema 1. (5 puntos)**

Un programa (P) se ejecuta en un computador cuya CPU (C1) funciona a una frecuencia de 2.4 GHz. El programa (P) contiene 50000 instrucciones estáticas, realiza  $2 \times 10^7$  operaciones de punto flotante y se ejecuta en 10 segundos en C1. La siguiente tabla muestra la distribución de instrucciones dinámicas y estáticas para el programa (P) junto con el CPI medio de cada tipo de instrucción.

	punto flotante	enteros	memoria
% instrucciones estáticas	10%	40%	50%
% instrucciones dinámicas	40%	35%	25%
CPI	6.0	2.0	6.5

a) **Calcula el rendimiento en MFLOPS del programa (P).**

b) **Calcula el CPI del programa (P).**

c) **Calcula el número de instrucciones dinámicas (N) del programa P.**

Se ha decidido que una rutina (R), que representa el 80% del tiempo de (P), sea programada en ensamblador.

d) **Calcula la ganancia en tiempo de ejecución (speedup) de la rutina (R) que se debería obtener respecto al compilador para que el programa (P) se ejecute 3 veces más rápida.**

Figura 15: Format d'examen corresponent al 1er Control del Curs 2017-2018 Q2

Tal i com s'ha comentat a l'apartat 2.5.1 d'aquest projecte, s'ha treballat amb dos formats d'exàmens diferents. Aquesta modificació s'ha produït en el canvi de quadrimestre, de mode que el 2n i 3er Control del Curs 2017-2018 del Q1 tenen el format de la Figura 14 mentre que el 1er Control del Curs 2017-2018 del Q2 té el format de la Figura 15.

2on Control Arquitectura de Computadors Curs 2017-2018 Q1

COGNOMS:

NOM:  DNI:

Figura 16: Graella corresponent a l'antic format emprat

1er Control Arquitectura de Computadors Curs 2017-2018 Q2

COGNOMS:

NOM:  DNI/NIE:

Figura 17: Graella corresponent a l'actual format emprat

Aquest canvi de format afecta principalment a la graella que els alumnes han d'omplir amb les seves dades, tal i com es pot apreciar a la Figura 16 i 17. Conjuntament, amb el professorat, es va acordar eliminar la petita graella situada a la part dreta de la Figura 16, per tal de poder fer més gran la graella principal. A més es va considerar oportú indicar en el camp del DNI, que també es podia escriure el NIE en cas de tractar-se d'un estudiant provinent de l'estranger, com es pot veure a la Figura 17. Aquest canvi va ser implementat degut a la gran quantitat d'alumnes estrangers que no omplien aquest camp. Cal remarcar que el format emprat actualment és el de la Figura 17.

## 4.2 Creació d'un document digitalitzat per a cada examen

Com s'ha indicat a l'apartat anterior, els arxius proporcionats pel professorat contenen tots els exàmens compresos en 3 o 4 PDFs. Per tal de poder analitzar a qui pertany cadascun dels exàmens és necessari separar aquests documents i generar un PDF per a cada examen. Aquesta funcionalitat s'encarregarà de realitzar-la l'eina generada per l'Eric Canals. Una vegada pujats els diferents arxius PDF, l'usuari haurà d'indicar de quantes pàgines està format l'examen. Aquesta automàticament generarà un PDF per a cada examen. Durant la realització d'aquest projecte s'ha emprat una eina online anomenada SmallPDF[15], la qual ens permet generar altres arxius PDF a partir del nostre document inicial.

## 4.3 Conversió dels documents PDF a imatges JPG

Per tal de poder treballar amb la llibreria OpenCV és necessari que el format d'entrada sigui JPG o PNG. Per aquest motiu és indispensable convertir els documents digitalitzats PDF a format JPG. Durant aquest procés només es generarà la imatge JPG de la primera plana del document digitalitzat, ja que serà la que contindrà la graella a processar. La imatge generada conservarà el mateix nom que l'arxiu PDF i serà creada en el mateix directori on hi figuri aquest.

Per a fer possible aquest conversió s'emprarà la llibreria Wand[14] per Python de ImageMagick. Aquesta llibreria ens permet fer conversions de format, editar i aplicar diferents transformacions en diferents tipus de formats d'imatges.

## 4.4 Identificació de les coordenades de la graella

El format de les imatges amb que es treballarà seran exàmens digitalitzats de l'assignatura d'Arquitectura de Computadors. Aquests contindran una graella formada per 50 rectangles repartits entre els diferents camps: Nom format per 15 rectangles, Cognom format per 26 rectangles i DNI format per 9 rectangles.

1er Control Arquitectura de Computadors Curs 2017-2018 Q2

COGNOMS:

NOM: 



 DNI/NIE:

Figura 18: Graella usada en els exàmens

L'objectiu d'aquesta etapa és arribar aconseguir extreure totes les coordenades dels rectangles que formen la graella situada a la part superior del examen. Cadascun d'aquests rectangles pot contenir o no un caràcter que forma part dels 3 camps que té la graella: Nom, Cognom i DNI. Per a fer-ho possible s'usaran diverses tècniques emprades a visió per computació. A continuació s'expliquen els diversos passos seguits per a l'extracció de les coordenades d'aquests camps.

#### 4.4.1 Suavització de la imatge

La gran majoria d'imatges escanejades presenten cert soroll digital. Aquesta variació aleatòria de la lluminositat o dels colors d'imatges digitals normalment és generat pel dispositiu d'entrada, en aquest cas, l'escàner de la impressora.

Aquest fenomen està acceptat en certes situacions, degut a la poca lluminositat o la excessiva amplificació d'una imatge, no obstant aquesta no és una situació en la que ens puguem permetre passar per alt aquest fenomen. El fet de tenir píxels amb valors aleatoris a la nostre imatge pot fer malbé més d'una predicció.

Per aquest motiu s'ha emprat un procediment molt conegut en el tractament d'imatges, anomenat desenfocament gaussià. Aquesta tècnica, mitjançant l'ús d'un algorisme, barreja lleugerament els colors dels píxels pròxims en un mapa de bits, el que provoca que la imatge perdi alguns detalls minúsculs i, d'aquesta manera, es vegi més suau. És un efecte àmpliament utilitzat en el programari de gràfics, generalment per reduir el soroll de la imatge i reduir els detalls.

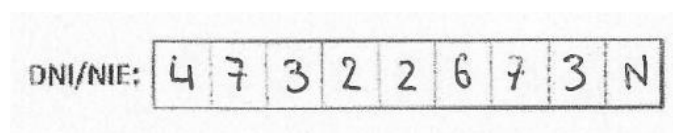


Figura 19: Imatge sense cap tipus d'efecte

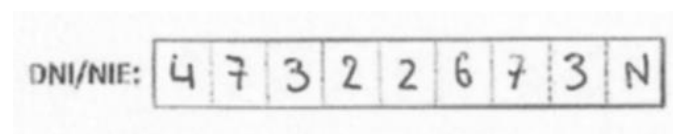


Figura 20: Imatge suavitzada mitjançant el desenfocament gaussià

#### 4.4.2 Binarització i retall de la imatge

Aquest procés consisteix en reduir la informació proporcionada per la imatge, per tal de facilitar el treball. D'aquesta manera la nostra imatge només tindrà valors compresos entre 0 (negre) i 255 (blanc).

Per tal de centrar-nos únicament en la part que volem estudiar, també es retallarà la imatge per la zona on es troben els camps: nom cognom i DNI compresa entre els píxels 120 i 330. D'aquest mode s'aconseguirà reduir el soroll que puguin provocar altres elements de l'examen en la nostra predicció.



Figura 21: Graella binaritzada

#### 4.4.3 Identificació dels contorns de la graella

Una vegada extreta aquesta imatge es procedirà a identificar les coordenades on està situada la graella. Per a fer-ho s'identificaran les línies que formen part del contorn de la taula. Primerament, es buscaran les línies horitzontals amb una llargada superior a 200 píxels mitjançant el procediment `getStructuringElement` de la llibreria OpenCV, el qual ens permet buscar estructures amb una morfologia i una mida mínima concreta.



Figura 22: Línies horitzontals de la graella

S'ha emprat el mateix procediment per a les línies verticals de la imatge amb una llargada superior a 55 píxels.



Figura 23: Línies verticals de la graella

Una vegada obtingudes ambdues imatges s'han superposat per a obtenir les coordenades del contorn de la taula.



Figura 24: Graella simplificada

Durant aquest procés d'identificació es va considerar l'opció de guardar les coordenades dels punts d'intersecció de la taula, tal i com es mostra a la Figura 25. Però a l'hora d'identificar aquests vèrtexs no es van obtenir els resultats esperats, ja que hi havia vèrtexs dels quals no va ser possible extreure les coordenades. Per aquest motiu es va optar per intentar reconèixer únicament el contorn de la graella principal, tal i com s'explica a l'apartat 4.4.5.



Figura 25: Vèrtexs de la graella simplificada

#### 4.4.4 Redimensionament de la imatge

Aquest procés consisteix en augmentar o reduir les dimensions d'una imatge. Les dimensions d'aquesta imatge es poden especificar manualment o simplement especificar un factor d'escalat. Dintre del redimensionament d'imatges hi ha diversos mètodes de interpolació, tot i que el més emprat per a realitzar el zoom d'una imatge és la interpolació bilineal.

La interpolació bilineal és una tècnica per calcular els valors d'una ubicació dintre d'una malla basada en cel·les de quadrícules properes. La diferència clau és que utilitza els quatre centres de les cel·les més properes.

Usant les quatre cel·les veïnes més properes, la interpolació bilineal assigna el valor de la cel·la de sortida prenent com a valor la mitjana ponderada. Aplica pesos basats en la distància dels quatre centres de cel·les més propers suavitzant la quadrícula de sortida de la trama.

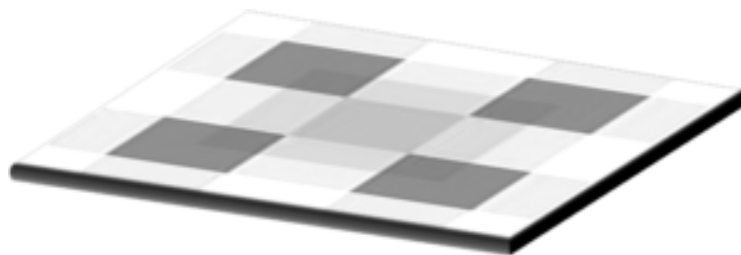


Figura 26: Interpolació bilineal

Aquesta etapa consisteix en aplicar un redimensionament adequat a la graella per a poder treballar fàcilment amb les caselles en les etapes posteriors. Tal i com s'explica més endavant, a l'hora d'identificar cadascuna de les caselles es divideix l'amplada total del camp entre el nombre de caselles que hi figuren. Per aquest motiu s'ha decidit aplicar un redimensionament de la graella que ens permeti dividir les caselles d'aquests camps sense haver d'usar decimals, és a dir, un nombre pròxim a les dimensions originals de la graella però múltiple de 27 (nombre màxim de caselles en l'eix horitzontal).

#### 4.4.5 Identificació dels contorns de cada camp

Una vegada obtingudes les coordenades del contorn de la graella i descartada l'opció de guardar les coordenades de les interseccions, es va procedir a identificar cadascun dels camps (nom, cognom i DNI) en diferents imatges. Per a realitzar aquest procediment, es van mesurar les dimensions que tenien les caselles de la graella estudiada. Es van extreure les següents dades: 60 píxels d'alçada i 46 píxels d'amplada. Sabent el contorn de la graella, el nombre de caselles que formen part de cadascun dels camps i les dimensions de cadascuna de les caselles es va poder separar fàcilment els tres camps que formaven part de la graella:



Figura 27: Camp on apareix el cognom de l'alumne



Figura 28: Camp on apareix el nom de l'alumne



Figura 29: Camp on apareix el DNI de l'alumne

#### 4.4.6 Identificació dels contorns de cada casella

Finalment, gràcies a les dimensions en píxels de cadascuna de les imatges s'ha pogut calcular les dimensions de cadascun dels rectangles que conté el caràcter. Per a realitzar aquest càlcul s'ha dividit l'amplada total de cadascun dels camps identificats anteriorment entre el nombre de caselles presents en aquests. D'aquest mode s'ha aconseguit obtenir una imatge per cadascuna de les caselles de la graella.



Figura 30: Rectangles contenidors dels caràcters

## 4.5 Tractament de la imatge final

Finalment, una vegada obtingudes les coordenades del rectangle on es troba cada caràcter és necessari el processat d'aquesta imatge final: retallar contorns per extreure'n imperfeccions (contorns de la graella), correcció de color, binarització de colors, redimensionament equitatiu de la imatge i identificació de caselles buides.

### 4.5.1 Retallat de la imatge

Una vegada obtingudes les coordenades del rectangle, caldrà acabar de polir les imperfeccions que presenta la imatge. En aquesta etapa el que es farà es retallar la imatge aplicant un marge de 0,1 en els 4 costats. Una vegada retallats aquests píxels es completaran aquests buit amb 4 marges negres de la mateixa mida. D'aquesta manera aconseguirem eliminar les imperfeccions provocades pel contorn de la graella.



Figura 31: Retallat d'imatge

### 4.5.2 Redimensionament equitatiu de la imatge

Fins el moment s'ha estat treballant amb les dimensions originals de les caselles. La nostra xarxa neuronal té un format d'entrada de 28x28 píxels. El problema és que les imatges extretes de la graella no són quadrades sinó rectangulars, és a dir, que no compten amb les mateixes dimensions pel que fa a l'alçada i l'amplada. De mode que si apliquem un redimensionament a aquestes imatges es produeix una deformació tal i com es pot apreciar a la Figura 32.

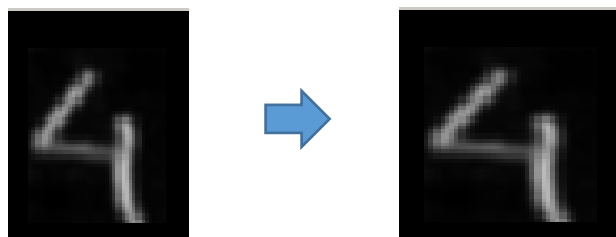


Figura 32: Deformació produïda pel redimensionament

Per a aquest fenomen no succeeixi cal convertir la imatge rectangular en una imatge quadrada abans d'aplicar-hi un redimensionament. Per aquest motiu s'ha emplenat el eix x de les imatges amb marges negres, fins a obtenir una imatge totalment quadrada. D'aquest mode al realitzar el redimensionament de la imatge mitjançant la llibreria OpenCV[9] no es deformarà la imatge, tal i com es pot apreciar a la Figura 33.



Figura 33: Redimensionament equitatiu de la imatge

#### 4.5.3 Correcció del color

Per tal d'intentar fer més semblants les imatges extretes dels exàmens amb les imatges dels jocs de proves, s'ha emprat la tècnica de correcció de color, ja que les imatges no acostumen a tindre suficient intensitat. Aquesta tècnica consisteix en tornar a calcular cadascun dels píxels respecte el valor màxim trobat a la imatge, és a dir, considerant que el valor màxim trobat és el 255, aplicar una regla de 3 a la resta de valors.

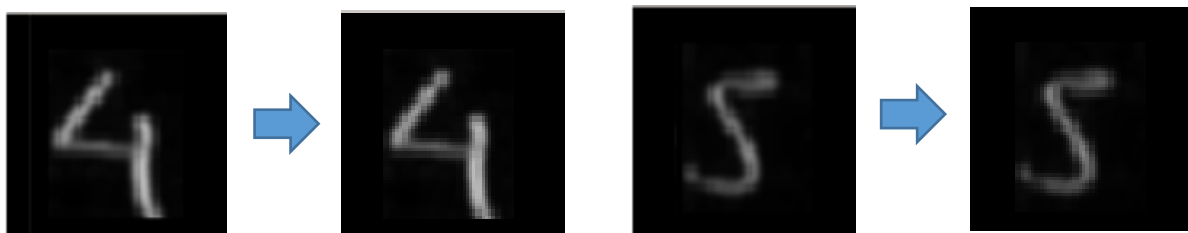


Figura 34: Correcció del color

#### 4.5.4 Binarització de colors

Les imatges dels jocs de proves posseeixen valors força contrastats, és a dir, la majoria de valors són pròxims a 0 o 255. Per tal de fer més similars les imatges obtingudes en els exàmens, s'ha realitzat una binarització de colors mitjançant un threshold òptim. Després de diverses proves s'ha arribat a la conclusió de que el threshold que genera un millor rendiment és el 40. Per tant, aquest serà el valor de tall per decidir si un color és blanc (255) o negre (0).



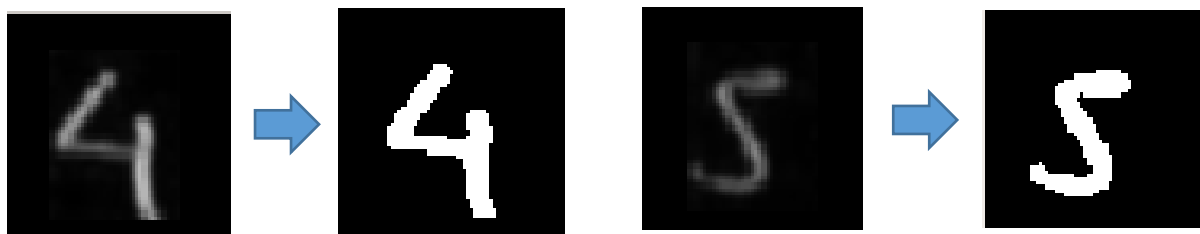


Figura 35: Binarització de colors

#### 4.5.5 Identificació de caselles buides

Una vegada ja es té la imatge processada i preparada per predir-ne el seu contingut és important identificar aquelles caselles que no contenen cap caràcter, ja que la xarxa neuronal no és capaç de diferenciar si hi ha algun caràcter o no. Per tant hem de ser nosaltres qui identifiqui si està buida o no. Per a fer-ho es recorrerà la imatge en busca de 20 píxels blancs. Si es passa aquest threshold es considerarà que la imatge no està buida i passarà a ser predita per la xarxa neuronal, en cas contrari es tractarà el següent caràcter.

## 4.6 Arquitectura de la xarxa neuronal

Una xarxa neuronal està constituïda per diferents nivells o capes que s'encarreguen d'extreure informació per tal d'inferir patrons entre aquelles entrades que presentin similituds. Tal i com es pot veure a la Figura 36 les xarxes neuronals presenten 3 tipus de capes: capa d'entrada, intermèdia o amagada i de sortida.

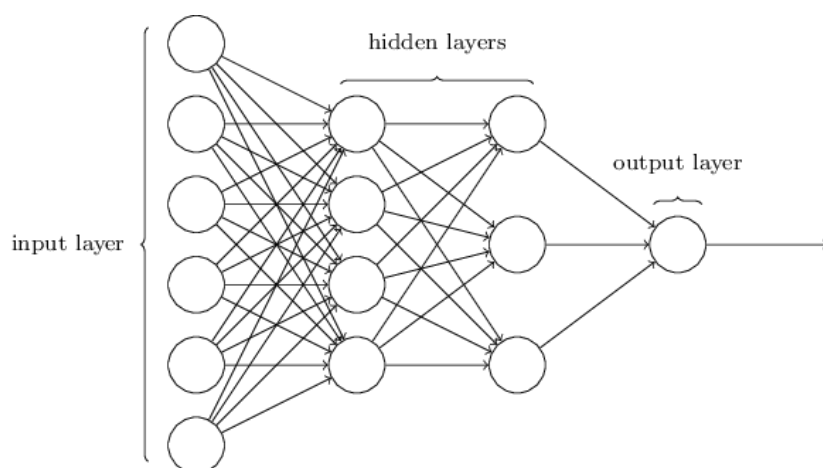


Figura 36: Estructura bàsica d'una xarxa neuronal

#### 4.6.1 Capa d'entrada

La capa d'entrada d'una xarxa neuronal és la que s'encarrega de recollir les dades i introduir-les dintre la xarxa neuronal. Cadascuna de les neurones que formen aquesta capa són anomenades neurones d'entrada.

### 4.6.2 Capa intermèdia o amagada

La capa intermèdia o amagada és aquella capa que es troba entre la capa d'entrada i sortida. En una xarxa neuronal poden haver-hi un nombre il·limitat de capes intermèdies. És l'única part de la xarxa neuronal que pot tenir més d'una instància. Aquestes capes són les encarregades d'ajustar els seus pesos i thresholds per tal d'acabar generant uns valors que acabin donant una predicció encertada a la capa de sortida. Cadascuna d'aquestes capes s'encarrega d'identificar uns certs patrons que es van repetint en les diferents entrades. A continuació s'expliquen algunes de les capes més comuns en el camp de reconeixement visual:

#### Capa convolució

La capa de convolució rep com a entrada o input la imatge i li aplica un filtre o nucli que ens torna un mapa de les característiques de la imatge original, d'aquesta manera aconseguim reduir la mida dels paràmetres. La convolució aprofita tres idees importants que poden ajudar a millorar qualsevol sistema de machine learning:

- Interaccions disperses: Al aplicar un filtre de menor grandària sobre l'entrada original podem reduir dràsticament la quantitat de paràmetres i càlculs.
- Paràmetres compartits, Al compartir els paràmetres entre els diferents tipus de filtres, s'ajuda a millorar l'eficiència del sistema.
- Representacions equivariants: Indiquen que si les entrades canvien, les sortides canviaran també de forma similar.

#### Capa pooling

La funció d'aquesta capa és reduir progressivament la mida espacial de la representació per reduir la quantitat de paràmetres i el temps de computació a la xarxa, és a dir, controlar l'execució excessiva. L'ús més comú és una capa d'agrupament amb filtres de mida 2x2. De manera que com es pot veure a la Figura 37 una matriu 4x4 queda reduïda en una matriu 2x2 on cadascuna de les caselles conté el valor màxim de les 4 caselles anteriors. Aquesta capa es sol usar després d'haver aplicat alguna capa de convolució per tal de reduir el temps d'execució.

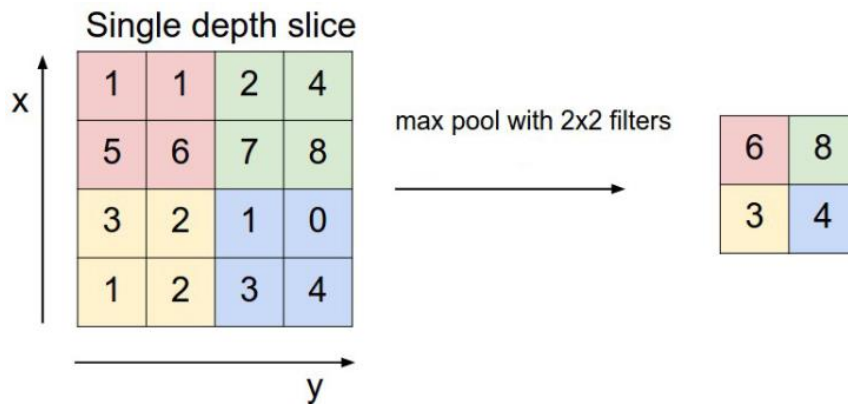


Figura 37: Reducció de la mida espacial realitzada per una capa de pooling 2x2

### Capa flatten

La capa flatten és l'encarregada de convertir tots els elements de dimensions 2 en un únic vector lineal. És utilitzada principalment després de les capes de convolució.

### Capa dropout

La capa de dropout serveix per minimitzar el problema de overfitting[16]. L'overfitting és el problema que sorgeix quan s'usen sempre les mateixes dades d'entrenament. Això provoca que tot i que el reconeixement de les dades d'entrenament sigui molt efectiu, alhora d'usar les dades reals, la nostra xarxa neuronal tingui certes mancances. Per tal d'evitar aquest problema, s'usen capes de dropout, les quals posen a 0 valors d'entrada aleatòriament. Aquestes capes només són actives durant la fase d'entrenament, durant la inferència no realitzen cap acció. L'únic paràmetre que requereixen és el percentatge de valors que volem posar a 0. La desavantatge del dropout és que fa que els models s'entrenin més lentament.

### Capa dense

La capa dense és l'encarregada de reduir la mida espacial de la representació en el nombre de sortides establert per l'usuari. Dintre d'aquesta capa es pot aplicar una funció d'activació. En aquest projecte s'han emprat dues: relu i sigmoide.

La funció d'activació relu aplica una rectificació als paràmetres d'entrada i està definida com:

$$f(x) = \max(0, x)$$

La funció d'activació sigmoide és molt emprada per descriure casos d'aprenentatge de sistemes complexos que mostren una progressió temporal. Està definida com:

$$f(x) = \frac{1}{1 + e^{-x}}$$

### 4.6.3 Capa de sortida

La capa de sortida d'una xarxa neuronal és la encarregada de extreure la informació generada per a la xarxa neuronal d'una manera entenedora. Cadascuna de les neurones que formen aquesta capa són anomenades neurones de sortida.

### 4.6.4 Models proposats

Durant el transcorregut del projecte s'han dissenyat, entrenat i provat diversos models per al reconeixement visual de caràcters escrits a mà. A continuació es mostren alguns models:

#### Primer model: Xarxa convolucional de 64x64 complexa

Aquest model va ser provat durant la fase inicial del projecte. Correspon al model dissenyat per l'Agustí Bau durant el seu Treball de Final de Grau a la FIB. Aquest model presentava un rendiment bastant bo tant en les dades d'entrenament com en les dades reals. El problema principal era la seva entrada, ja que era de 64x64. Aquest fet implicava que entrenar el model fos molt costós, de mode que es va desestimar. A continuació, es poden veure les capes emprades en aquest model:

- Capa d'entrada de 64x64
- Capa de convolució amb 64 kernels 3x3.
- Capa de convolució amb 64 kernels 3x3.
- Capa de Max pooling 2x2
- Capa de convolució amb 128 kernels 3x3
- Capa de convolució amb 128 kernels 3x3
- Capa de Max pooling 2x2
- Capa flatten
- Capa dense de 128 amb funció d'activació relu
- Capa dropout 0.5
- Capa dense de 64 amb funció d'activació relu
- Capa dropout 0.5
- Capa dense de 10 sortides per a la xarxa de dígit i 26 per a la xarxa de lletres amb funció d'activació softmax

#### Segon model: Xarxa convolucional de 28x28 senzilla

Aquest model va ser usat durant l'inici del projecte. Va ser el model amb el que es va començar a realitzar les primeres proves: entrenaments, avaluacions... Tot i presentar un rendiment molt bo amb les dades d'entrenament, a l'usar les dades reals va sofrir una davallada important. Finalment, es va considerar que era un model massa senzill per a realitzar prediccions de caràcters escrits a mà. A continuació, es poden veure les capes emprades en aquest model:

- Capa d'entrada de 28x28
- Capa de convolució amb 30 kernels de 5x5
- Capa Max pooling 2x2
- Capa de convolució amb 15 kernels de 3x3

- Capa de Max pooling 2x2
- Capa Flatten
- Capa dropout 0.2
- Capa dense de 128 amb funció d'activació relu
- Capa dense de 50 amb funció d'activació relu
- Capa dense de 10 sortides per a la xarxa de dígit i 26 per a la xarxa de lletres amb funció d'activació softmax

### Tercer model: Xarxa convolucional de 28x28 complexa

Aquest model va ser usat durant la part intermèdia del projecte. Tot i ser un model complexa, no es va obtenir el rendiment esperat durant la identificació de caràcters. A continuació, es poden veure les capes emprades en aquest model:

- Capa d'entrada de 28x28
- Capa de convolució amb 32 kernels de 5x5
- Capa Max pooling 2x2
- Capa de convolució amb 64 kernels de 3x3
- Capa de Max pooling 2x2
- Capa de convolució amb 128 kernels de 3x3
- Capa de Max pooling 2x2
- Capa flatten
- Capa dropout 0.5
- Capa dense de 512 amb funció d'activació relu
- Capa dense de 256 amb funció d'activació relu
- Capa dense de 10 sortides per a la xarxa de dígit i 26 per a la xarxa de lletres amb funció d'activació sigmoide

### Quart model: Xarxa convolucional de 64x64 complexa adaptada a 28x28

Finalment, després de diverses proves es va optar per acabar usant el model proposat en el Treball de Final de Grau[5] d'en Agustí Bau. Tot i això s'ha modificat per satisfer les necessitats d'aquests projecte. Alguns d'aquests canvis són l'ús d'una capa d'entrada de 28 bits i una capa de sortida amb la funció d'activació sigmoide. A continuació es mostra el model final emprat en aquest projecte:

- Capa d'entrada de 28x28
- Capa de convolució amb 64 kernels 3x3.
- Capa de convolució amb 64 kernels 3x3.
- Capa de Max pooling 2x2
- Capa de convolució amb 128 kernels 3x3
- Capa de convolució amb 128 kernels 3x3
- Capa de Max pooling 2x2
- Capa flatten
- Capa dense de 128 amb funció d'activació relu
- Capa dropout 0.5
- Capa dense de 64 amb funció d'activació relu
- Capa dropout 0.5

- Capa dense de 10 sortides per a la xarxa de dígit i 26 per a la xarxa de lletres amb funció d'activació sigmoide

## 4.7 Entrenament de la xarxa neuronal

En aquesta part del projecte s'hauran d'entrenar les xarxes neuronals dissenyades anteriorment. Per tal de dur a terme aquest propòsit s'exposarà ambdues xarxes a dos jocs de proves: MNIST i EMNIST.

### 4.7.1 MNIST

Aquest joc de proves serà l'escollit per entrenar la xarxa neuronal encarregada del reconeixement de caràcters numèrics. MNIST engloba 60.000 entrades d'imatges de dígit escrits a mà per a l'entrenament de la xarxa i 10.000 més per a l'avaluació. Keras[8] permet importar aquest joc de dades directament per la qual cosa no serà necessari descarregar-se cap arxiu per a poder entrenar la nostra xarxa neuronal. La Figura 38 mostra alguns dels exemples que podem trobar en el MNSIT.

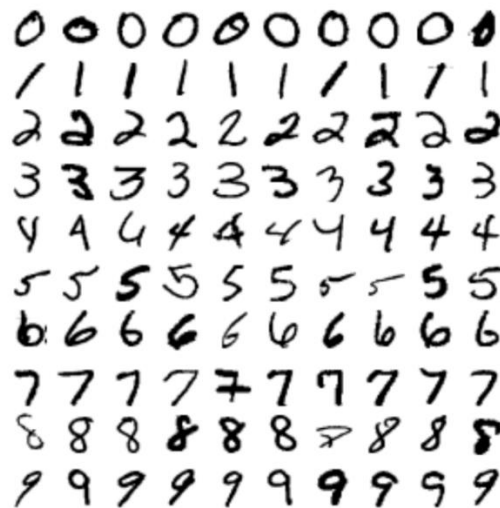


Figura 38: Nombres continguts en el MNIST

### 4.7.2 EMNIST

Aquest joc de proves serà l'escollit per entrenar la xarxa neuronal encarregada del reconeixement de caràcters alfabètics. EMNIST està format per diversos jocs de proves més petits que contenen diversos tipus de caràcters: dígit, alfabètics, etc. El problema dels jocs de dades alfabètics és que contenen tant majúscules com minúscules, per la qual cosa s'ha hagut de filtrar aquest joc de dades a l'hora d'entrenar la nostra xarxa neuronal, ja que només es volen considerar lletres majúscules. Finalment ha quedat un joc de proves format per 188958 imatges de caràcters alfabètics en majúscules per a l'entrenament i 31346 imatges per a l'avaluació. La Figura 39 mostra alguns dels caràcters que podem trobar en el EMNSIT.

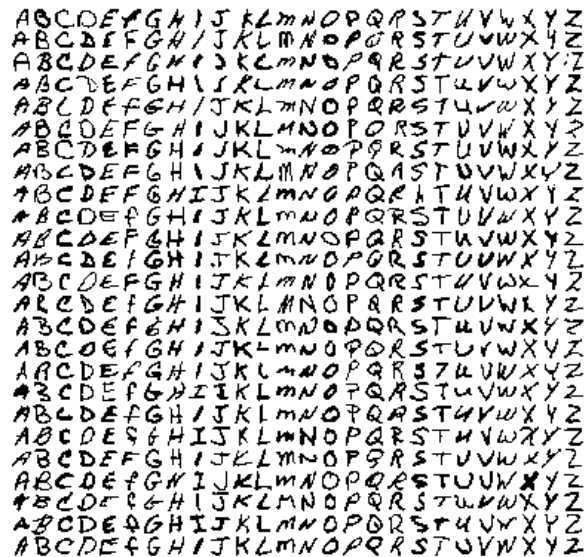


Figura 39: Caràcters continguts en el EMNIST

### 4.7.3 Augmentació de dades

Les imatges emprades per a l'entrenament de les xarxes neuronals estan molt estandarditzades, ja que no presenten cap tipus de rotació, translació, etc. Per aquest motiu, durant l'entrenament de les xarxes neuronals s'ha aplicat una tècnica anomenada augmentació de dades. Aquest procediment consisteix en modificar els jocs de proves MNIST i EMNIST aplicant-hi rotacions, augmentacions, translacions etc. D'aquesta manera s'ha aconseguit apropar més aquests jocs de proves estàndard cap a les imatges dels exàmens i incrementar el percentatge d'encerts en les prediccions realitzades per les xarxes neuronals.

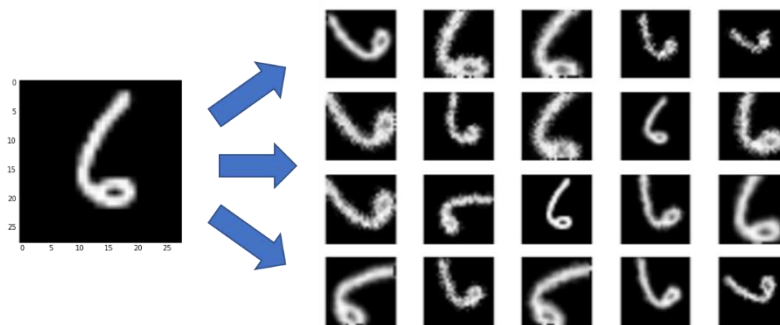


Figura 40: Aplicació de la augmentació de dades sobre una imatge

L'aplicació d'aquesta tècnica sobre el joc de dades d'entrenament ha estat un fet crucial en l'èxit d'aquest projecte. Durant la fase intermèdia del projecte la xarxa neuronal seleccionada no era capaç de reconèixer amb precisió els nombres 0, 1, 6. Al començar a aplicar les diferents transformacions geomètriques al joc de dades d'entrenament, aquest percentatge d'encerts va incrementar de forma considerable.

## 4.8 Avaluació de la xarxa neuronal

Per tal de realitzar l'avaluació d'ambdues xarxes neuronals s'ha hagut d'analitzar el contingut real escrit de cadascun dels exàmens, manualment, per tal de veure si el nom i cognom de l'examen era realment el que figurava al llistat. Una vegada realitzat aquest anàlisi s'ha hagut de documentar tota aquesta informació per tal de poder comparar la predicció de la xarxa neuronal amb el contingut que realment figurava a l'examen. Després de realitzar aquest procediment s'han obtingut els següents resultats:

- Xarxa neuronal encarregada del reconeixement de dígit: 88,91% d'encerts amb les dades reals
- Xarxa neuronal encarregada del reconeixement de lletres: 81,03% d'encerts amb les dades reals

## 4.9 Estructura de la predicció

Una vegada proporcionada la predicció de la xarxa neuronal obtenim per a cada examen analitzat una estructura amb els següents atributs:

- numberDNI: Conté la part numèrica predita del DNI
- predictionNumberDNI: Conté per cada nombre del DNI les diferents probabilitats de tots els dígit possibles.
- letterDNI: Conté la part alfabètica predita del DNI
- predictionLetterDNI: Conté les diferents probabilitats de tots caràcters alfabètics per a la casella on hi figura la lletra del DNI.
- name: Conté el nom predit per la xarxa neuronal
- predictionName: Conté per cada lletra les diferents probabilitats de tots els caràcters alfabètics possibles.
- surname: Conté el cognom predit per la xarxa neuronal
- predictionSurname: Conté per cada lletra les diferents probabilitats de tots els caràcters alfabètics possibles.
- realFileName: Conté el nom del arxiu associat a l'examen
- realDNI: Aquest camp indica el DNI real de l'alumne. No s'emplenarà fins que l'algoritme hagi estat capaç d'identificar l'alumne.
- realName: Aquest camp indica el nom real de l'alumne. No s'emplenarà fins que l'algoritme hagi estat capaç d'identificar l'alumne.
- realSurname: Aquest camp indica el cognom real de l'alumne. No s'emplenarà fins que l'algoritme hagi estat capaç d'identificar l'alumne.
- realCsvIndex: Aquest camp indica el nombre del llistat associat a l'alumne. No s'emplenarà fins que l'algoritme hagi estat capaç d'identificar l'alumne.
- possibleCsvIndex: Aquest camp conté els possibles alumnes propietaris d'aquests examen. No s'emplenarà fins que es comenci a executar l'algoritme d'identificació.

## 4.10 Algoritmes d'identificació

Per tal de identificar les prediccions obtingudes, amb els alumnes que figuren al llistat, caldrà usar diferents algoritmes d'identificació que permetin un cert interval d'error en la predicció.



Una vegada s'associï un alumne a un examen aquest quedarà descartat del llistat. A continuació s'exposen els diversos algoritmes emprats en l'ordre d'utilització.

#### 4.10.1 Associació directa

Aquest algoritme d'identificació és el més restrictiu. Consisteix en calcular la lletra del DNI mitjançant l'algoritme conegut com mòdul 23. L'algoritme consisteix en aplicar l'operació aritmètica de mòdul 23 al nombre del DNI. El resultat és un nombre comprès entre el 0 i el 22. En relació amb la Taula 5 s'assigna una lletra.

Residu	0	1	2	3	4	5	6	7	8	9	10	11	12
Lletra	T	R	W	A	G	M	Y	F	P	D	X	B	N

Residu	13	14	15	16	17	18	19	20	21	22
Lletra	J	Z	S	Q	V	H	L	C	K	E

Taula 5: Taula d'assignació de la lletra del DNI

El càlcul d'aquesta lletra es realitza amb els nombres predits per la xarxa neuronal. De manera que si la lletra calculada amb els nombres predits i la lletra del DNI predita coincideixen es considera un hit. Si el DNI predit apareix en el llistat s'associa l'alumne del llistat a l'examen.

#### 4.10.2 Associació directa amb un error

Aquest algoritme d'identificació permet un error en la predicció del DNI. A diferència de l'algoritme anterior no és té en compte que el càlcul de la lletra del DNI coincideixi amb la lletra predita. Donada una predicció d'un examen es busca al llistat tots aquells candidats que presenten un únic error amb el DNI predit. En cas de que únicament hi hagi un únic candidat es produeix una assignació directa.

#### 4.10.3 Associació d'exàmens que presenten més d'un error

Aquest algoritme engloba tots aquells exàmens que contenen un nombre especificat d'errors en el DNI major que 1. Aquest nombre estipulat d'errors s'especificarà en la crida de la funció, juntament amb el threshold d'error permès en el nom i el cognom predits. L'algoritme proposat està pensat per anar-se executant seqüencialment, mentre es van incrementant el nombre d'errors permesos i acotant els thresholds. Donada la predicció d'un examen es busca al llistat tots aquells candidats que presenten un nombre menor o igual al numero estipulat d'errors. Una vegada es tenen identificats els possibles candidats es passarà a mesurar la quantitat d'error que presenta cadascun dels candidats respecte a la predicció.

Per tal de mesurar aquesta quantitat s'ha creat un altre algoritme encarregat de comparar les diferents modificacions que s'haurien de realitzar al nom i cognom predits per acabar sent iguals a cadascun dels possibles candidats. Aquest algoritme pondera cadascuna de les modificacions amb els següents valors:

- Intercanvi d'una lletra en la predicció: 0.25
- Inserció d'una lletra en la predicció: 0.5
- Eliminació d'una lletra en la predicció: 1

Aquestes dues ponderacions són idèntiques alhora de comparar el nom i el cognom. D'aquesta manera l'algoritme proposat ens retornarà dues distàncies:

- $distNom = 0,25 \times Intercanvis + 0,5 \times Insercions + 1 \times Eliminacions$
- $distCognom = 0,25 \times Intercanvis + 0,5 \times Insercions + 1 \times Eliminacions$

Una vegada obtingudes ambdues distàncies s'hauran de comparar amb el threshold fixat anteriorment. Si diversos candidats superen els thresholds establerts s'escollirà aquell candidat que presenti una menor quantitat d'error en ambdós indicadors.

La eficàcia d'aquest algoritme està en anar augmentant el nombre d'errors mentre es redueixen els thresholds permesos. D'aquesta manera el que s'aconsegueix és anar reduint el nombre d'exàmens no identificats a mesura que es permeten més fallades. Cal remarcar que la relació entre el nombre d'errors permesos i thresholds és bastant conservadora. A continuació es mostra una taula amb els diferents paràmetres emprats en l'algoritme:

Nombre d'errors	Threshold nom	Threshold cognom
2	1	2
3	1	2
4	1	2
5	0.75	1
6	0.75	1
7	0.5	1.5
8	0.5	1.5
9	0.5	1.5

Taula 6: Taula amb els paràmetres de l'algoritme d'associació d'exàmens

## 4.11 Entrada i sortida

### 4.11.1 Entrada

L'eina elaborada en aquest projecte espera un cert format d'entrada per a funcionar correctament. Aquesta entrada consisteix en la ruta d'un directori on hi ha de figurar el llistat en format csv i els diferents exàmens digitalitzats, tal i com es mostra a la Figura 41 i 42.

```
$ python identifyMarcelNewSheetPDF.py testPDFNewSheet/
```

Figura 41: Comanda per executar l'eina de reconeixement d'exàmens

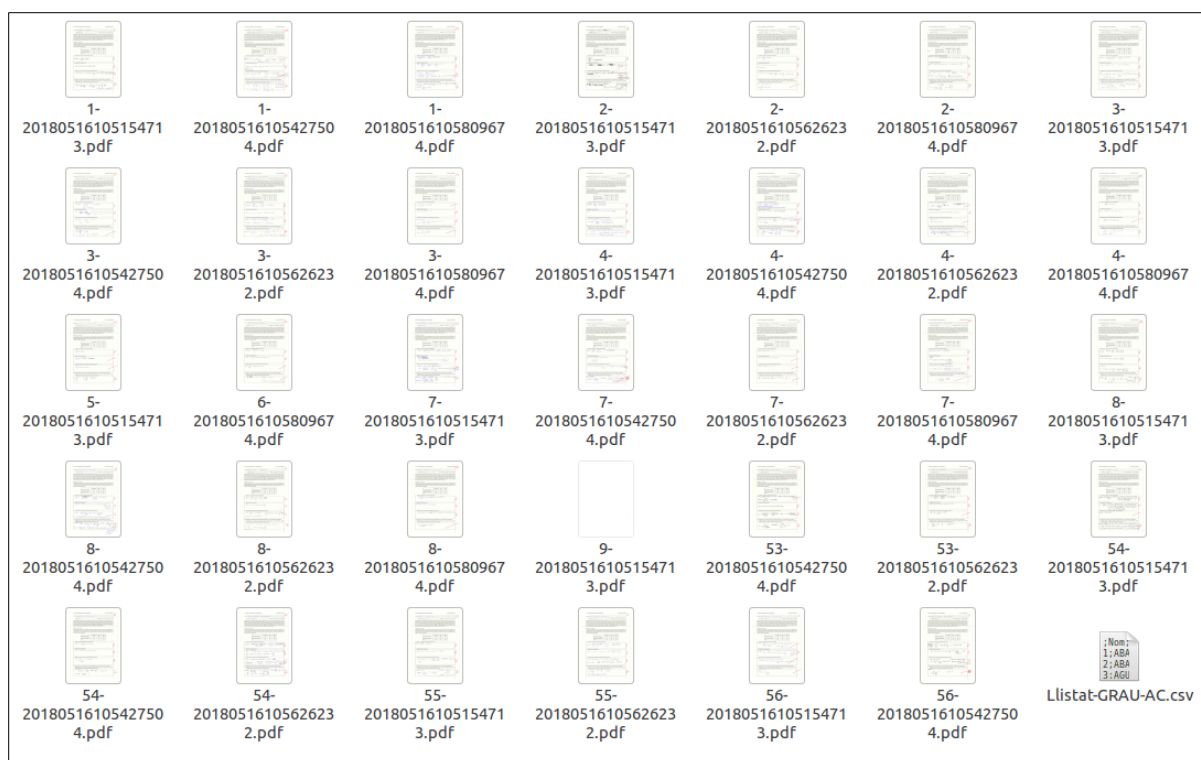


Figura 42: Contingut del directori d'entrada

El llistat Csv d'entrada conté les dades personals de tots els alumnes matriculats a la assignatura d'Arquitectura de Computadors. Per a que la eina pugui llegir el contingut d'aquest llistat és necessari que respecti la següent estructura: número de la fila en el llistat, nom complet de l'alumne, DNI sense lletra i DNI amb lletra, tal i com es pot apreciar a la Figura 43.

```
;Nom;DNI;Passaport;
1;ARROYO PINAR, MARCEL;47322673;47322673N
2;PÉREZ PÉREZ, LUIS;47322672;47322672O
3;SÁNCHEZ LÓPEZ, CHRISTIAN;47322671;47322671I
4;IBARRA GARCIA, CARLOS;47322670;47322670Q
5;SARABIA BENÍTEZ, MARCOS;47322669;47322669R
```

Figura 43: Contingut del llistat d'entrada csv

Aquest format d'entrada ha estat escollit en consens per tal de facilitar al professorat la generació d'aquest llistat, ja que es possible generar-lo automàticament amb la informació emmagatzemada de l'alumnat matriculat a la assignatura d'Arquitectura de Computadors, tal i com s'explica en l'apartat 3.3.2

#### 4.11.2 Sortida

Tal i com s'ha explicat en l'apartat 1.5 aquest projecte no compta amb una interfície gràfica. Per aquest motiu es vol proporcionar un format de sortida estàndard per a facilitar la utilització d'aquestes dades en possibles projectes futurs que puguin proporcionar un entorn gràfic. Aquesta eina genera un llistat Csv molt similar al d'entrada, anomenat Output.csv. A continuació es mostra el format de sortida generat per a l'eina:

```
csvNumber;Nom;DNI;Passaport;PDFName  
1;ARROYO PINAR, MARCEL;47322673;47322673N;1-20180516105154713.pdf  
2;PÉREZ PÉREZ, LUIS;47322672;473226720;2-20180516105154713.pdf  
3;SÁNCHEZ LÓPEZ, CHRISTIAN;47322671;47322671P;3-20180516105154713.pdf  
4;IBARRA GARCIA, CARLOS;47322670;47322670Q;4-20180516105154713.pdf  
5;SARABIA BENÍTEZ, MARCOS;47322669;47322669R;None|
```

Figura 44: Contingut del llistat de sortida del csv

L'arxiu generat per a l'eina de reconeixement conté les dades de tots els alumnes matriculats a l'assignatura incloses en el llistat d'entrada juntament amb un nou camp anomenat PDFName. En aquesta columna hi figura el nom de l'arxiu que conté l'examen digitalitzat associat a un alumne. Els alumnes que no tinguin un examen assignat tindran un "None" a la columna PDFName. Aquest fet pot ser degut a que la eina no hagi estat capaç de reconèixer amb prou seguretat a l'alumne com a propietari del examen o bé perquè aquest no s'hagi presentat a l'examen.

## 5 Resultats

Aquest apartat pretén exportar els resultats obtinguts al llarg del projecte dels diferents models emprats per al reconeixement de caràcters escrits a mà, vistos en l'apartat 4.6.4. Cal considerar que tant per al reconeixement de dígit com el de caràcters alfabètics s'ha emprat la mateixa xarxa neuronal. Únicament varien en la capa de sortida. A continuació es mostren els resultats obtinguts amb els diferents models proposats en aquest projecte.

Per tal de reduir els noms de les xarxes neuronals en les gràfiques s'ha usat la següent nomenclatura:

- M1: Xarxa convolucional de 64x64 complexa
- M2: Xarxa convolucional de 28x28 senzilla
- M3: Xarxa convolucional de 28x28 complexa
- M4: Xarxa convolucional de 64x64 complexa adaptada a 28x28

El primer indicador a analitzar és el percentatge d'encerts amb les dades d'entrenament. Aquestes dades provenen dels resultats obtinguts amb el joc de proves MNIST (dades numèriques) i EMNIST (dades alfabètiques). Tal i com es pot observar les prediccions realitzades sobre el joc de proves numèric són més acurades que les prediccions sobre el joc de proves alfabètic. Això és degut a que la xarxa neuronal encarregada de predir dades numèriques té 10 possibles sortides mentre que l'encarregada de predir caràcters alfabètics en té 26. Aquesta diferència entre les possibles sortides d'ambdues xarxes neuronals fa que la probabilitat d'error en el cas de la xarxa neuronal encarregada de predir caràcters alfabètics sigui més gran.

A la Figura 45 es pot observar que hi hagut una evolució ascendent respecte els percentatges dels diferents models emprats al llarg del projecte. El model amb el percentatge d'encerts més elevat és el M4 amb un 99,44% d'encerts en dades numèriques i 98,24% d'encerts en dades alfabètiques. El percentatge d'encert en dades d'entrenament ens pot donar una aproximació sobre com de precís pot ser el nostre model.

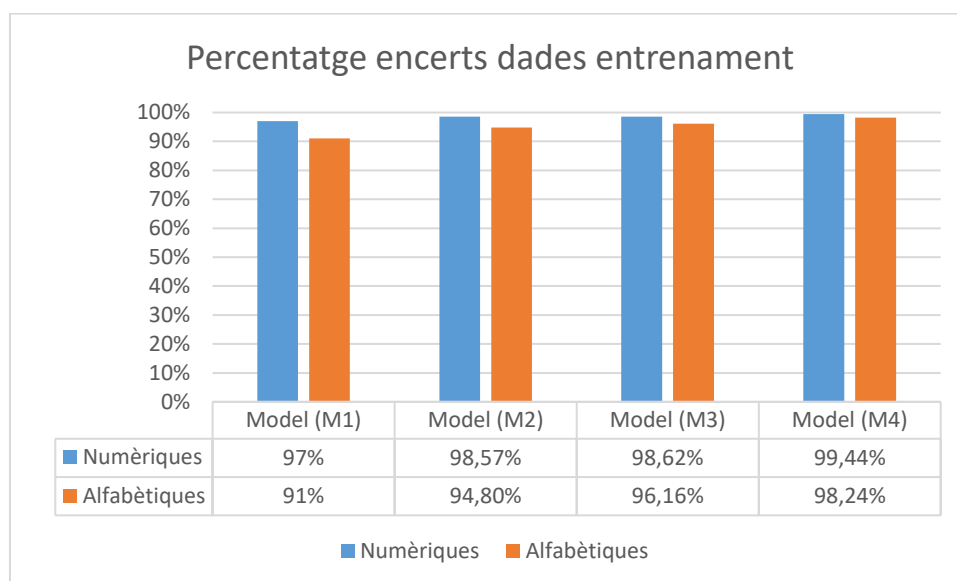
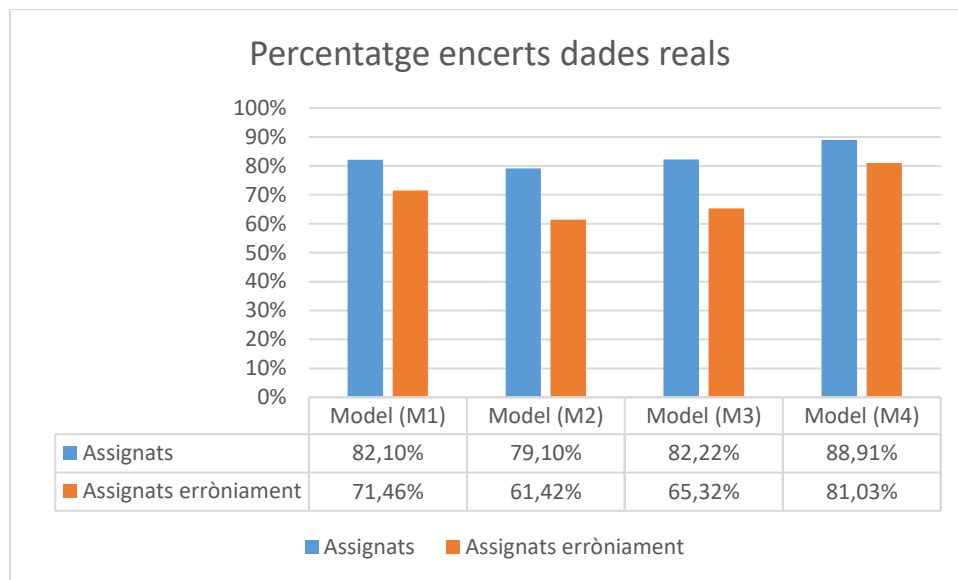


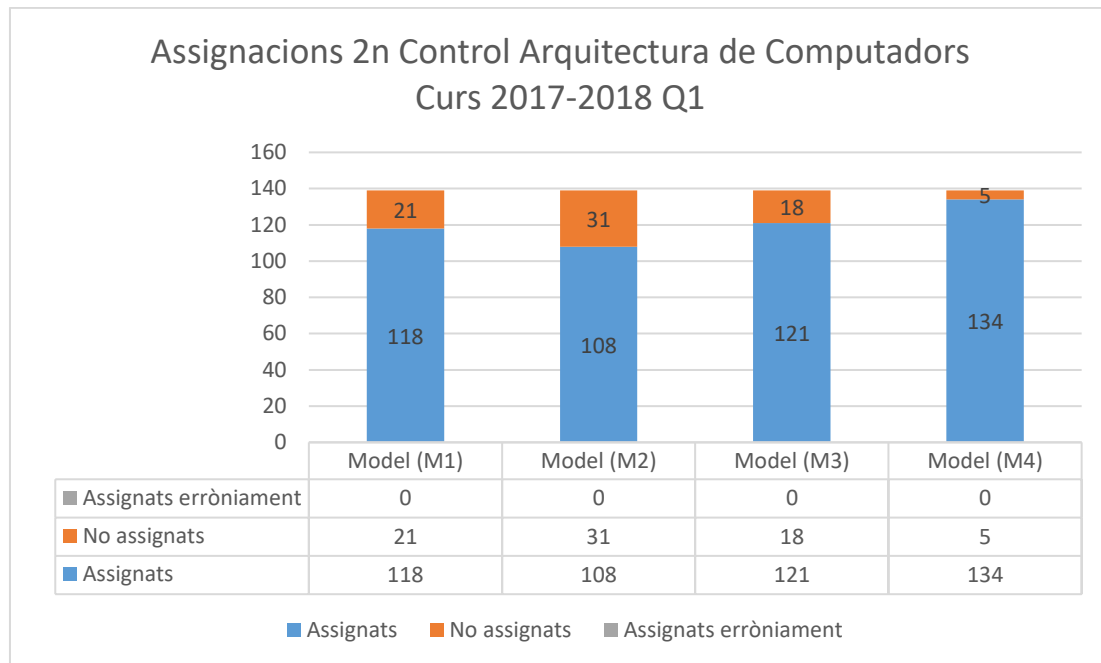
Figura 45: Gràfica del percentatge d'encerts amb dades d'entrenament

El segon indicador a analitzar és el percentatge d'encerts amb les dades reals. Aquestes dades provenen dels resultats obtinguts en el 2n Control d'Arquitectura de Computadors Curs 2017-2018 Q1. Per tal d'obtenir aquests resultats s'ha hagut d'afegir manualment la sortida esperada per a cada predicció. Tal i com s'ha comentat en la gràfica anterior els percentatges d'encerts entre el model per a lletres i números difereixen força. En aquest cas, la diferencia entre ambdós models amb el mateix disseny es fa molt més notòria. Això és degut a la qualitat de les imatges recollides, ja que no estan tant estandarditzades com les dels jocs de proves. En el joc de dades reals ens podem trobar dígitos que ni tan sols una persona sigui capaç d'identificar. Tot i això el model M4 ha aconseguit uns percentatges d'encerts molt bons: 88,91% en dades numèriques i 81,03% en dades alfabètiques tal i com es pot veure a la Figura 46.



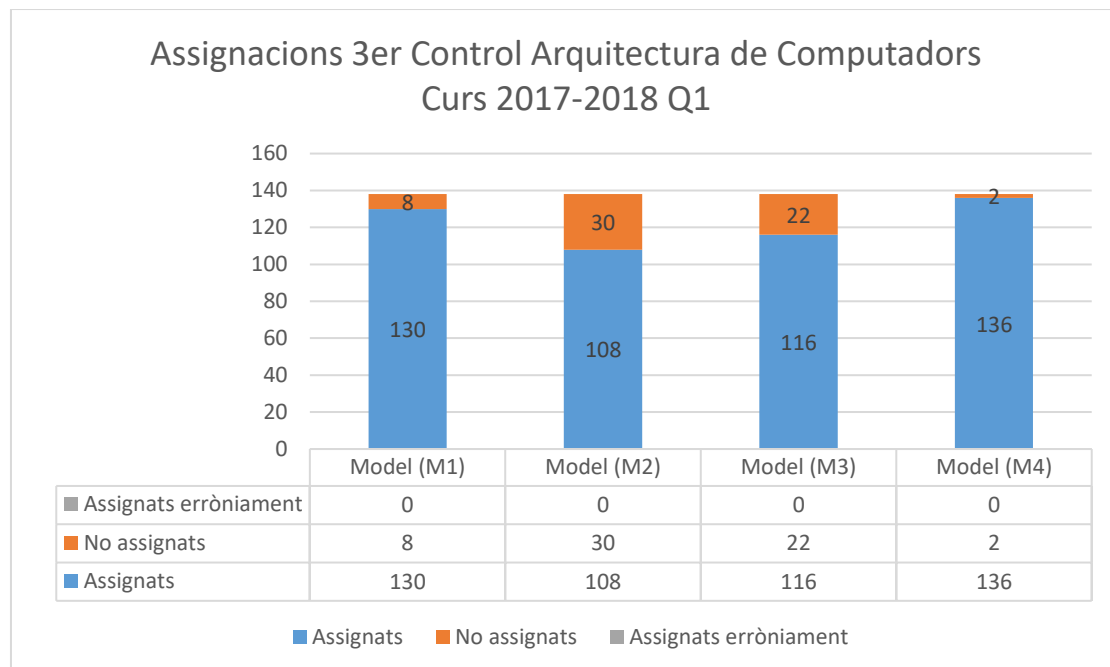
**Figura 46: Gràfica del percentatge d'encerts amb dades reals**

El tercer indicador a analitzar recull el nombre d'exàmens assignats correctament, erròniament i no assignats en el 2n Control d'Arquitectura de Computadors Curs 2017-2108 Q1. Tal i com es pot apreciar a la imatge el nombre d'exàmens assignats es correspon amb el percentatge d'encerts de la gràfica anterior, els quals si que són uns indicadors fiables de la qualitat de la nostra xarxa neuronal. Mitjançant l'ús del model M4 s'ha aconseguit assignar correctament 134 exàmens dels 139 possibles. La resta no han pogut ser assignats (5 exàmens). El percentatge d'assignació correcta del model M4 és d'un 96,40% mentre que el percentatge del model M1, corresponent al Treball de Final de Grau d'en Agustí Bau, és de 84,48% tal i com es pot veure a la Figura 47. Cal remarcar que totes aquestes assignacions s'han realitzat mitjançant els algorismes d'identificació desenvolupats en aquest projecte. En el cas del model M1, l'ús d'aquests algorismes ha potenciat la seva fiabilitat, ja que no s'ha assignat cap examen erròniament, mentre que en el Treball Final de Grau de l'Agustí Bau s'assignaven 5 exàmens incorrectament. La resta de models tampoc presenten errades per la qual cosa podem dir que els nostres algorismes d'identificació són fiables.



**Figura 47: Gràfica d'assignacions d'exàmens 2n Control Curs 2017-2018 Q1**

El quart indicador a analitzar recull el nombre d'exàmens assignats correctament, erròniament i no assignats en el *3er Control d'Arquitectura de Computadors Curs 2017-2108 Q1*. Tal i com es pot apreciar a la imatge el nombre d'exàmens assignats es correspon amb el percentatges d'encerts de la Figura 46, els quals si que són uns indicadors fiables de la qualitat de la nostra xarxa neuronal. Mitjançant l'ús del model M4 s'ha aconseguit assignar correctament 136 exàmens dels 138 possibles. La resta no han pogut ser assignats (2 exàmens). El percentatge d'assignació correcta del model M4 és d'un 98,55% mentre que el del model M1 es queda amb un percentatge de 94,2% tal i com es pot veure a la Figura 48. En aquesta avaluació tampoc s'ha assignat cap examen erròniament, per la qual cosa es remarca l'eficiència dels algorismes d'identificació.



**Figura 48: Gràfica d'assignacions d'exàmens 3er Control Curs 2017-2018 Q1**

El cinquè indicador analitzat recull el nombre d'exàmens assignats correctament, erròniament i no assignats en el *3er Control d'Arquitectura de Computadors Curs 2017-2018 Q1*. Aquest examen va ser realitzat per 209 alumnes. Mitjançant l'ús del model M4 s'ha aconseguit assignar correctament 196 exàmens. La resta d'exàmens no han pogut ser assignats (13 exàmens). En aquesta avaluació el model M1 pateix una davallada important, ja que baixa fins un percentatge d'assignació correcta de 71,29% tal i com es pot veure a la Figura 49. El model M4 es queda amb un percentatge de 93,77% d'assignacions correctes. Aquest valor és un resultat molt bo ja que tot i el gran nombre d'exàmens a associar conserva un percentatge força bo. Com en els tres casos anteriors, en aquesta avaluació tampoc s'ha assignat cap examen erròniament, per la qual cosa podem concloure amb que els algoritmes d'identificació tenen una fiabilitat alta. Cal destacar que el temps d'execució de l'eina de reconeixement visual elaborada amb aquest joc de proves, el qual és el més extens fins a dia d'avui, té un temps d'execució de 20 minuts i 40 segons.



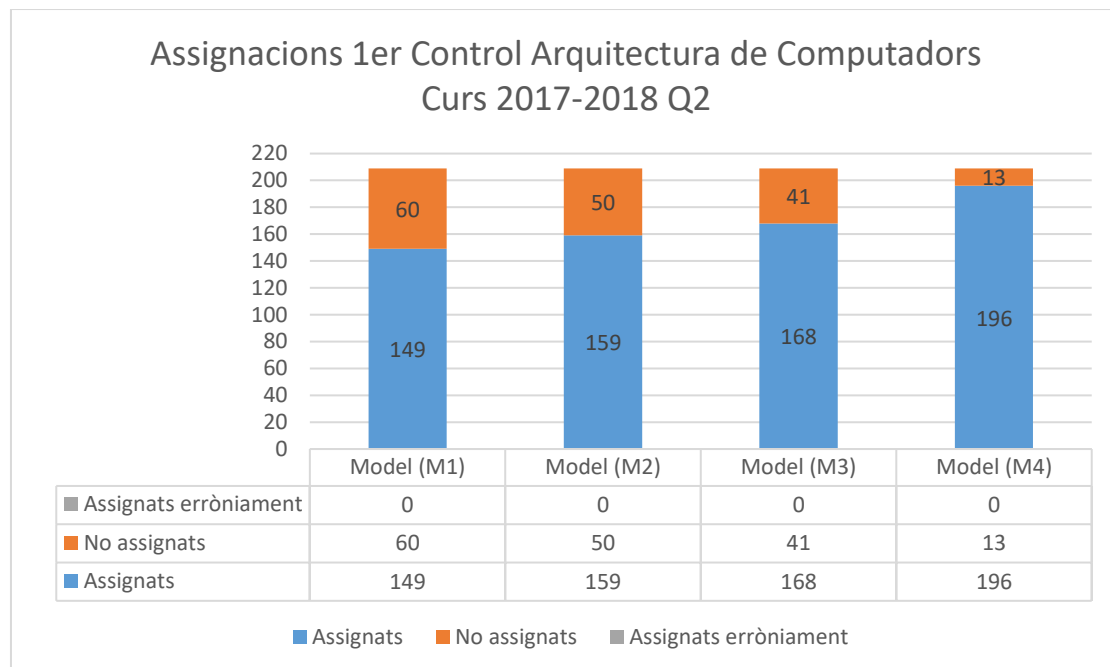


Figura 49: Gràfica d'assignacions d'exàmens 1er Control Curs 2017-2018 Q2

Durant les assignacions del *3er Control d'Arquitectura de Computadors Curs 2017-2108 Q1* amb el model M4, es va analitzar el nombre d'assignacions d'exàmens que es realitzaven en cadascun dels algoritmes d'identificació explicats en l'apartat 4.10. s'ha considerat interessant compartir aquestes dades per tal de reflectir la importància que tenen en aquest projecte. A la Figura 50 podem veure l'evolució de les assignacions d'exàmens a mesura que es van executant els diferents algoritmes.

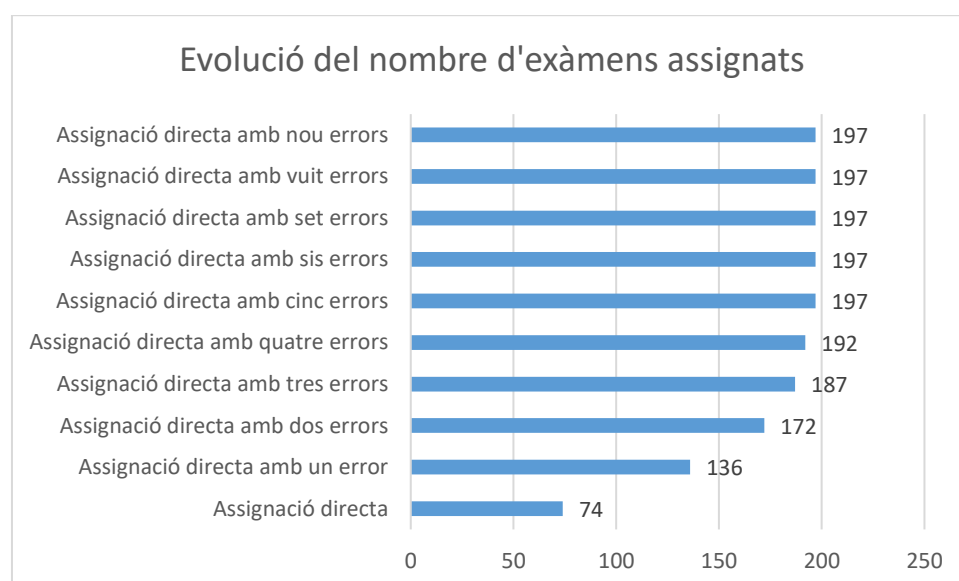


Figura 50: Gràfica de l'evolució del nombre d'exàmens assignats

L'últim indicador a analitzar és el temps d'entrenament requerit per a cada xarxa neuronal. El temps del model M1 no hi figura ja que no s'ha aconseguit reproduir el procés d'entrenament, degut a la seva elevada càrrega computacional i la nostre limitació de dispositius hardware. Per tal de realitzar l'entrenament dels diversos models s'han emprat els jocs de proves MNIST i EMNIST.

La Figura 51 mostra una diferencia poc significant entre l'entrenament de caràcters alfabètics o numèrics dintre del mateix model. No obstant, hi ha una gran diferència entre el temps d'entrenament del model M4 amb la resta de models. Això es degut a que el model M4 és el més complex de tots ells. El temps d'entrenament del model M4 corresponent a la xarxa neuronal de caràcters alfabètics és de 95,46 minuts mentre la de caràcters numèrics és de 92,96 minuts.

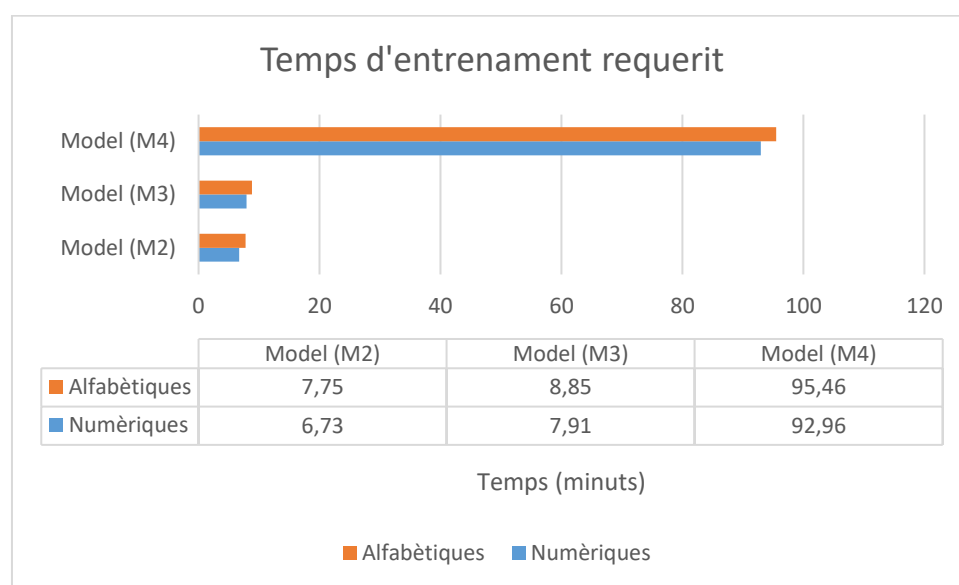


Figura 51: Gràfica temps d'entrenament requerit

## 6 Conclusions i treball futur

### 6.1 Conclusions

L'objectiu principal d'aquest projecte era dissenyar una eina capaç de associar un examen digitalitzat al seu corresponent propietari. Després de veure els resultats obtinguts en el nostre model final (M4) podem afirmar que s'ha assolit l'objectiu fixat inicialment, ja que els percentatges d'assignació correcta en les 3 avaluacions analitzades són molt bons.

L'èxit d'aquest projecte es basa en la combinació de les àrees emprades per a l'elaboració d'aquesta eina: processat d'imatge, machine learning i correcció d'errors. El processat d'imatge usat en aquest projecte ens ha permès maximitzar la informació a extreure d'una imatge i modificar-la per a facilitar la comprensió de la xarxa neuronal. El machine learning ens ha permès identificar tant caràcters alfabètics com numèrics escrits a mà en una imatge amb un percentatge d'encerts força bo. Per últim, la correcció d'errors ens ha permès incrementar el percentatge d'encerts fent de la nostra eina un producte consistent i eficaç.

Cal tenir en compte que la nostra eina està lligada a limitacions que impedeixen l'objectiu d'aconseguir un 100% d'encerts en l'associació d'exàmens amb els seus respectius alumnes. La majoria d'aquestes limitacions radiquen d'un mal ús de les indicacions especificades a l'inici de l'examen, les quals descriuen que com s'ha d'omplir cadascun dels camps de la graella.

Algunes d'aquestes limitacions analitzades serien: garagots en les caselles a analitzar, cal·ligrafia excessivament cursiva, abreviacions en noms compostos, manca del segon nom en la graella, inclusió o supressió d'una "i" entre el primer i segon cognom, lletra situada en la primera posició del NIE...

Tot i que algunes d'aquestes limitacions es podrien solucionar en projectes futurs, tant els percentatges de correcta assignació com els de incorrecta assignació fan que la eina compleixi amb les necessitats plantejades inicialment per al professorat de l'assignatura d'Arquitectura de Computadors.

Per concloure, podem dir que s'ha aconseguit una eina fiable i eficaç per a l'associació d'exàmens amb moltes possibilitats de créixer en un futur. Actualment, la nostra eina s'usa únicament per a l'assignatura d'Arquitectura de Computadors però podria ser adoptada per a qualsevol assignatura que adopti el format d'examen vist en aquest projecte.

### 6.2 Treball futur

Aquest apartat recull totes aquelles millores que es podrien aplicar en un futur per a millorar les limitacions exposades anteriorment o simplement per millorar el rendiment d'aquesta implementació. A continuació s'expliquen algunes d'aquestes propostes:

Especificar de forma més acurada el contingut esperat a la graella

Tot i que l'explicació que hi figura en els exàmens de l'assignatura d'Arquitectura de Computadors descriu detalladament el contingut esperat, hi continuen havent-hi alumnes que

no posen el seu nom complert, realitzen abreviatures en els noms compostos, empren una cal·ligrafia excessivament cursiva o escriuen els caràcters a sobre de la graella.

Per aquest motiu, es considera oportú revisar el text descriptiu encarregat d'explicar als alumnes el contingut esperat a la graella, ja que moltes d'aquestes errades són les que fan reduir el percentatge d'encerts.

### Implementar diverses millores en el tractament d'errors

En aquest projecte s'han implementat diversos algorismes de corregir cert marge d'error en les nostres prediccions. D'aquest mode s'ha aconseguir incrementar considerablement el percentatge d'encerts. Tot i això, hi ha diverses tècniques que s'haguessin pogut aplicar si s'hagués disposat de més temps:

- **Detecció de la primera lletra del NIE:** Durant els últims anys s'ha notat un increment d'alumnes estrangers matriculats a la FIB. Els estrangers que resideixen a Espanya tenen un document d'identificació diferent al DNI. Aquest document és el NIE el qual conté un número d'identificació compostat per un caràcter alfabètic (X, Y o Z), seguit de 7 nombres i un caràcter alfabètic de control. Actualment el procediment que es segueix en el procés d'identificació del DNI és el següent: primeres 8 caselles predicció sota la xarxa neuronal de caràcters numèrics i última casella predicció sota la xarxa neuronal de caràcters alfabètics. Degut aquest increment d'alumnes estrangers es podria sotmetre a la primera casella del camp del DNI sota una predicció d'ambdues xarxes neuronals. En cas de que la xarxa neuronal de caràcters alfabètics identifiqués la lletra X, Y o Z, corresponents a la primera lletra del NIE, es podria considerar aquest alumne com a candidat a tindre un NIE, en comptes d'un DNI.
- **Inclusió o supressió d'una "i" entre el primer i segon cognom:** Durant les proves realitzades amb els diferents models s'ha detectat que alguns dels alumnes inserien una "i" entre els seus cognoms quan aquesta no hi apareixia en el llistat. També s'ha identificat el cas contrari. Per tal de solucionar aquest conflicte, es podria intentar detectar si al predir la lletra "i" tant a l'esquerre com a la dreta d'aquesta casella hi figura una casella buida. Aquest procediment es podria aplicar també al llistat d'alumnes comparant si al costat d'una lletra "i" hi figuren espais en blanc.
- **Ús de la capa sigmoide:** Un dels motius pels quals es va decidir implementar la capa sigmoide en el nostre model era per realitzar un tractament d'errors sobre la nostra predicció. De mode que si no érem capaços d'associar un alumne amb una predicció fóssim capaços d'elaborar una nova predicció amb els percentatges dels altres caràcters associats a una casella. Finalment aquest procediment no s'ha adoptat degut als bons resultats aconseguits amb el tractament d'errors ja implementat.

### Construcció de l'entorn mitjançant un contenidor

Per tal de facilitar l'ús d'aquesta eina, es podria realitzar la construcció d'un contenidor que permetés emprar l'entorn requerit en aquest projecte sense necessitat d'instal·lar les diferents llibreries de Python.

Per a la realització d'aquesta tasca s'ha pensat en l'ús de contenidors Docker[17], els quals ens permeten construir un entorn amb les seves respectives llibreries de manera còmode i senzilla.

# Referències

- [1] Marvin Minsky, Seymour A. Papert, Léon Bottou. Perceptrons: An Introduction to Computational Geometry. [Consulta: 4 març 2018]. Disponible a: <[https://books.google.es/books/about/Perceptrons.html?id=PLQ5DwAAQBAJ&source=kp\\_cover&redir\\_esc=y](https://books.google.es/books/about/Perceptrons.html?id=PLQ5DwAAQBAJ&source=kp_cover&redir_esc=y)>
- [2] ABBY. ABBYY FineReader 14. [Consulta: 4 març 2018]. Disponible a: <<https://www.abbyy.com/en-eu/finereader/>>
- [3] Nuance. Nuance's OmniPage 18. [Consulta: 4 març 2018]. Disponible a: <<https://www.nuance.com/en-gb/print-capture-and-pdf-solutions/optical-character-recognition/>>
- [4] Recogniform Technologies. Recogniform Desktop Reader. [Consulta: 4 març 2018]. Disponible a: <<http://www.recogniform.net/eng/recogniform-desktop-reader.html>>
- [5] Bau Pericón, A. Eina d'ajuda a la correcció d'exàmens [privat]. Treball final de grau, UPC, Facultat d'Informàtica de Barcelona. [Consulta: 26 febrer 2018].
- [6] Yann LeCun, Corinna Cortes, Christopher J.C. Burges. The MNIST Database. [Consulta: 4 març 2018]. Disponible a: <<https://www.nist.gov/itl/iad/image-group/emnist-dataset>>
- [7] NIST. EMNIST. Data actualització: 5 abril 2017 [Consulta: 4 març 2018]. Disponible a: <<https://www.nist.gov/itl/iad/image-group/emnist-dataset>>
- [8] Wikipedia. Scrum (software development) - wikipedia, the free encyclopedia. [Consulta: 3 juny 2018]. Disponible a: <[https://en.wikipedia.org/wiki/Scrum\\_\(software\\_development\)](https://en.wikipedia.org/wiki/Scrum_(software_development))>
- [9] Keras Documentation. Keras: The Python Deep Learning library. [Consulta: 5 març 2018]. Disponible a: <<https://keras.io/>>
- [10] OpenCV. [Consulta: 5 març 2018]. Disponible a: <<https://opencv.org/>>
- [11] TensorFlow. [Consulta: 5 març 2018]. Disponible a: <<https://www.tensorflow.org/>>
- [12] Michael Nielsen. *Neural Networks and Deep Learning*. [Consulta: 10 març 2018]. Disponible a: <<http://neuralnetworksanddeeplearning.com/index.html>>
- [13] Jason Brownlee. *Machine Learning Mastery*. [Consulta: 10 març 2018]. Disponible a: <<https://machinelearningmastery.com>>
- [14] SmallPDF. [Consulta: 6 juny 2018]. Disponible a: <<https://smallpdf.com/es>>
- [15] Wand. [Consulta: 3 juny 2018]. Disponible a: <<https://pypi.org/project/Wand/>>

[16] Overfitting and underfitting with machine learning algorithms. [Consulta 3 juny 2018]. Disponible a : <<https://machinelearningmastery.com/overfitting-and-underfitting-with-machine-learning-algorithms/>>

[17] Docker. What is a docker? [Consulta: 3 juny 2018]. Disponible a: <<https://www.docker.com/what-docker>>

# Apèndix

Degut a l'elevat volum de codi generat en aquest projecte s'ha considerat adjuntar l'enllaç del repositori de Github.

Enllaç: <https://github.com/marcelarroyo/tfg>